REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
01-12-2004

2. REPORT TYPE
Bi-annual Performance/Technical Report

3. DATES COVERED *(From - To)*
06/01/2004 - 11/30/2004

4. TITLE AND SUBTITLE
Bi-annual (06/2003--11/2004) Performance/Technical Report
for ONR YIP Award under Grant N00014-03-1-0466 Interval Type-2
Energy Efficient Wireless Sensor Networks Using Fuzzy Logic Systems
Lifetime Analysis

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014 – 03 –1 –0466

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Liang, Qilian

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Texas at Arlington
Office of Sponsored Projects
PO Box 19145
Arlington, TX 76019

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution is Unlimited.

20041223 010

13. SUPPLEMENTARY NOTES

14. ABSTRACT

During the period of 6/1/2004 -- 11/30/2004, we have proposed different approaches on energy efficient wireless sensor networks. (1) We applied interval type-2 fuzzy logic systems to wireless sensor network lifetime analysis. (2) A distributed sensor deployment scheme using fuzzy logic systems was proposed. (3) We proposed a new approach for sensed signal strength forecasting in wireless sensors using interval type-2 fuzzy logic system. (4) Energy and delay aware packet transmission in wireless sensor networks was studied. (5) An asynchronous contention-based and energy-efficient MAC protocol (ACEMAC) for was proposed for wireless sensor networks. Seven papers were produced during the past six months, and are attached to this report.

15. SUBJECT TERMS
Wireless Sensor Network, Energy Efficiency, Fuzzy Logic, Network Lifetime.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Qilian Liang
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			
U	U	U	UU	80	**19b. TELEPHONE NUMBER** *(Include area code)* 817-272-1339

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Wireless Sensor Network Lifetime Analysis Using Interval Type-2 Fuzzy Logic Systems

Haining Shu and Qilian Liang

Department of Electrical Engineering

University of Texas at Arlington

Arlington, TX 76019-0016 USA

E-mail: shu@wcn.uta.edu, liang@uta.edu

Abstract

Prolonging the lifetime of energy constrained wireless sensor networks is a crucial challenge in sensor network research. In this paper, we present a novel approach based on fuzzy logic theory to analyze the lifetime performance of a sensor network. We demonstrate that a type-2 fuzzy membership function(MF), i.e., a Gaussian MF with uncertain variance is most appropriate to model a single node lifetime in wireless sensor networks. In our research, we concern two basic sensor placement schemes: square-grid and hex-grid. Two fuzzy logic systems(FLSs): a singleton type-1 FLS and an interval type-2 FLS are designed to perform lifetime estimation of the entire sensor network. Simulation results show that the interval type-2 FLS in which the antecedent membership functions are modeled as type-2 fuzzy sets outperforms the singleton type-1 FLS when the single node lifetime behaves the nature of Gaussian MFs with uncertain standard deviation.

1 Introduction

Wireless sensor network represents significant advantages over conventional personnel-rich methods of long term data collection and monitoring. Starting in the 1980s, networked microsensors technology has been widely used in military applications. Examples of these applications include the Cooperative Engagement Capability (CEC), Remote Battlefield Sensor System (REMBASS), Advanced Deployable System (ADS) and Tactical Remote Sensor System (TRSS) [1]. Although a majority of military applications deploy sensor nodes randomly, a number of other applications require the placement of sensors at desired locations. Such placement-friendly sensor networks are developed rapidly for infrastructure security, environment and habitat monitoring, traffic control etc [1]. A real-world 32 node habitat monitoring sensor network system was deployed on a small island off the coast of Maine to study nesting patterns of Petrels [2]. Other applications involve the use of sensors in buildings for environmental monitoring which may include chemical sensing and detection of moisture problems. Structural monitoring [3] and inventory control are some other applications of such networks.

Wireless sensor network consists of certain amount of small and energy constrained nodes. Basic components of sensor node include a single or multiple sensor modules, a wireless transmitter-receiver module, a computational module and a power supply module. Such networks are normally deployed for data collection where human intervention after deployment, to recharge or replace node batteries may not be feasible, resulting in limited network lifetime. Most applications have pre-specified lifetime requirements, for instance the application in [2] has a lifetime requirement of at least 9 months. Thus estimation of lifetime of such networks prior to deployment becomes a necessity. Prior works on evaluating lifetime have considered networks where sensor nodes are randomly deployed. [4] gives the upper bound on lifetime that any network with the specified number of randomly deployed nodes, source behavior and energy can reach while [6] discusses the upper bounds on lifetime of networks with cooperative cell based strategies. Network liftime of fixed deployment schemes are recently studied in [7]. Jain and Liang [7] discovers that in wireless sensor networks, a single node lifetime behaves the nature of normal distribution which brings the first light of exploring the network lifetime behavior given the knowledge of nodes lifetimes.

In this paper, we deal with the issue of lifetime analysis and estimation for wireless sensor networks in which the sensor nodes are deployed at desired locations. We propose to apply interval type-2 fuzzy logic systems (FLSs) for network lifetime analysis and estimation. Our approach is entirely different from all prior research. We demonstrate that a type-2 fuzzy membership function(MF), i.e., a Gaussian MF with uncertain variance is most appropriate to model a single node lifetime in wireless sensor networks. Two fuzzy logic systems(FLSs): a singleton type-1 FLS and an interval type-2 FLS are constructed to perform lifetime estimation of the entire sensor network. In our work, we concern two basic sensor placement schemes: square-grid and hex-grid. We believe that these two schemes can serve as basis for evaluating more complex schemes for their lifetime performance prior to deployment and help justify the deployment costs.

The rest of this paper is organized as follows. In section 2, we detail two sensor deployment schemes used for this study and the basic concepts on network coverage, connectivity and lifetime. Section **??** gives an overview of interval type-2 fuzzy logic systmes. In section 4, we demonstrate that a single node lifetime can be modeled with Gaussian MFs. We found that a type-2 Gaussian MF with uncertain variance is most appropriate to model a single node lifetime in wireless sensor networks. We apply this knowledge and design an interval type-2 FLS in section 5 to analyze network lifetime. A singleton type-1 FLS is constructed as well for performance comparison. Simulation results and discussions are also presented in section 5. Section 6 concludes this paper.

2 Preliminaries

2.1 Basic Model and Assumptions

Consider identical wireless sensor nodes placed in a square sensor field of area A. All nodes are deployed with equal energy. Each sensor is capable of sensing events up to a radius r_s, the sensing range. Communication range r_c is defined as the distance beyond which the transmitted signal is received with signal to noise ratio (SNR) below the acceptable threshold level. In this paper, We assume the communication range r_c to be equal to the sensing range r_s.

Direct communication between two sensor nodes is possible only if their distance of separation r is such that $r \leq r_c$. We call such nodes *neighbors.* Communication between a sensor node and its non-neighboring node is achieved via peer-to-peer communication. Thus the maximum allowable distance between two nodes who wish to communicate directly is $r_{max} = r_c = r_s$. A network is said to be deployed with minimum density when the distance between its neighboring nodes is $r = r_{max}$.

2.2 Placement Schemes

The simplest placement schemes involve regular placement of nodes such that each node in the network has the same number of neighbors. We arrive at two basic placement schemes by considering cases where each sensor nodes has four and three neighbors. This leads us to the *square-grid* and *hex-grid* placement schemes shown in Fig. 1(a) and (b) respectively.

A sensor node placement scheme that uses two neighbors per sensor node has been described in [13]. We believe that these three elementary placement schemes can serve as basis for other placement schemes, because a placement scheme of any complexity can be decomposed into two-neighbor, three-neighbor and four-neighbor groups. Both grids shown have the same number of nodes[1] and nodes in both grids are equidistant from their respective neighbors (with distance of separation r).

2.3 Coverage and Connectivity

Coverage and connectivity are two important performance metrics of networks and hence a discussion on them becomes imperative before the lifetime of the network can be defined.

Coverage scales the adequacy with which the network covers the sensor field. A sensor with sensing range r_s is said to cover or sense a circular region of radius r_s around it. If every point in the sensor field is within distance r_s from at least one sensor node, then the network is said to provide complete or 100% coverage.

Various levels of coverage are acceptable depending on the application. In critical applications, complete coverage is required at all times. Any loss of coverage leads to a sensing gap in the field.

[1] The *Hex-grid* has lower density than the *Square-grid.* With 36 nodes deployed, the network with *Square-grid* covers an area of $25r^2$, and the *Hex-grid* covers an area of $48r^2$, almost double that of the *square grid.*

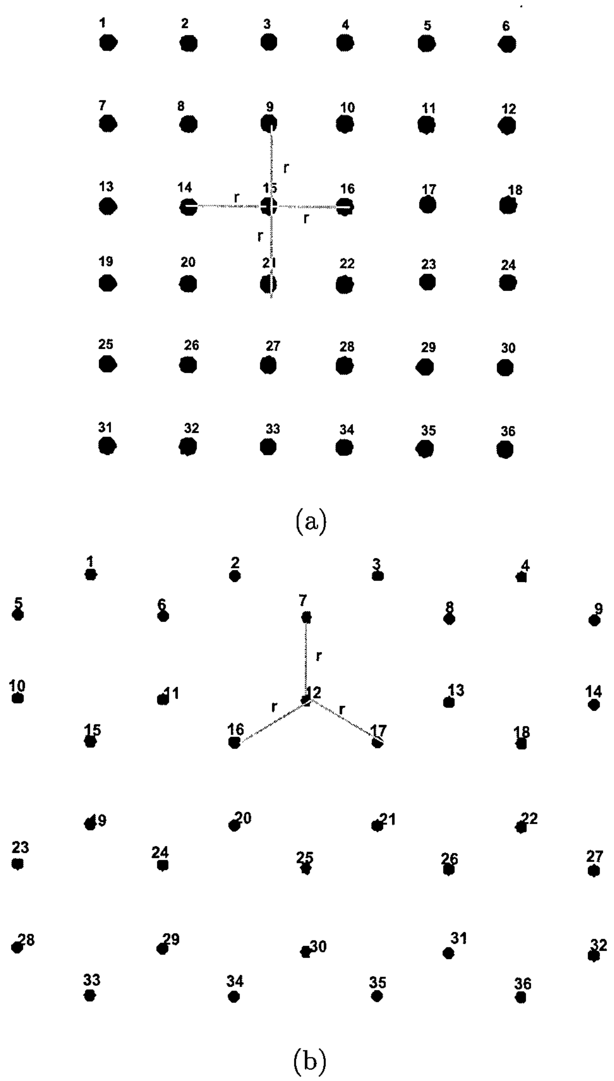


Figure 1: Placement Schemes for a 36 node sensor network: (a) Square-Grid (b) Hex-Grid.

Such gaps cause breach of security in case of surveillance applications. Also, in applications which require data with high precision , a sensing gap leads to inaccuracies. For such networks any loss of coverage renders the network nonfunctional. In some other applications a small loss of coverage may be acceptable.

Connectivity scales the adequacy with which nodes are able to communicate with their peers. One of the strengths of sensor networks arises from their ability to aggregate data collected from different sensor nodes. This requires adequate communication between sensor nodes. Any node should be able to communicate with any other node for proper functioning of the network. If a single node gets isolated due to failure of all its neighbors, it will be unable to communicate with the rest of the network. If a large number of nodes fail due to lack of energy, a part of the network may get completely disconnected from the rest. If a large number of nodes fail due to lack of energy, a part of the network may get completely disconnected from the rest. In our analysis we assume that only 100% connectivity is acceptable and the network fails with any loss of connectivity. An example of a sensor placement scheme that concentrates mainly on coverage as its parameter of interest can be found in [14], where a sensor placement algorithm for grid coverage has been proposed.

In our analysis we require the network to provide complete coverage and connectivity. We give equal importance to both parameters and declare the network nonfunctional if either of them falls below their desired levels.

2.4 Lifetime

The basic definition of lifetime, or more precisely the post-deployment active lifetime of a network is the time measured from deployment until network failure. Based on the levels of coverage and connectivity required to deem a network functional, network failure can be interpreted in different ways. Since only complete coverage and connectivity are acceptable to us, network failure corresponds to the first loss of coverage or connectivity.

In this paper, we concentrate on finding the minimum lifetime of a network, the worst case scenario. To be able to evaluate this minimum lifetime, we need to know the lifetime of a single sensor node the, the minimum number of node failures that cause network failure, and the positional relationship [2] between the failed nodes.

[2] Positional relationship between two nodes can be that the two nodes are diagonal, adjacent or completely unrelated.

2.4.1 Minimum Density Networks

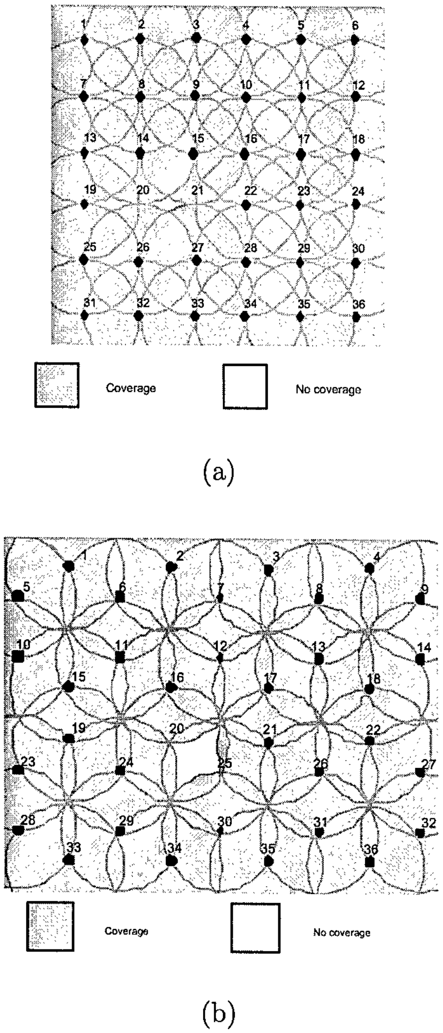
Consider the square-grid and the hex-grid deployed with minimum density. Both schemes survive the failure of a single node without loss of either connectivity or coverage implying that the minimum number of node failures that can lead to network failure is greater than one. Failure of any two neighboring nodes causes loss of coverage and hence network failure as indicated in Figs 2(a) and (b).

Thus the minimum number of node failures that cause network failure is two and these two nodes must be adjacent to each other (neighbors). A network may undergo multiple node failures and still be connected and covered if any of the failed nodes are not neighbors. But the absolute minimum number of node failures that can cause network failure is two.

2.4.2 Network lifetime for dense placement

Let N_{min} be the number of sensor nodes deployed in a sensor field of area A with minimum density. In a minimum density network, the separation between neighbors is r_{max}, the maximum range of the sensor nodes. A network deployed with higher density would require a larger number of nodes N, $N > N_{min}$ and smaller separation between neighboring nodes r, $r < r_{max}$. Since the range of a sensor node r_{max} is greater than the distance to its immediately adjacent nodes, it is now be able to communicate not only with the nodes immediately adjacent to it, but also with other nodes which fall in its range. In other words, each node now has more number of neighbors that any node in a minimum density network.

We discussed earlier that the minimum lifetime of a square-grid or hex-grid network with $r = r_{max}$(minimum density) is the time to failure of two neighboring nodes. High density networks can lose $(N - N_{min})$ nodes and still be deployed with minimum density and fail only after the further loss of a minimum of two neighboring nodes. Since the network after $(N - N_{min})$ node failures is deployed with minimum density, the time taken for the failure of any two neighboring nodes would be equal to the minimum lifetime of a minimum density network (T_{md}) which was discussed in the



(a)

(b)

Figure 2: Loss of coverage due to failure of two neighboring nodes: (a)Square-grid: Failure of nodes 20 and 21 causes loss of coverage. (b)Hex-grid: Failure of nodes 20 and 25 causes loss of coverage.

preceding section. If t_{dense}, the time taken for the failure of $(N - N_{min})$ nodes can be found, we define the lifetime of a high density network (T_{hd}) as:

$$T_{\text{hd}} = t_{\text{dense}} + T_{\text{md}} \tag{1}$$

3 Introduction to Type-2 Fuzzy Sets and Interval Type-2 Fuzzy Logic Systems

3.1 Introduction to Type-2 Fuzzy Sets

The concept of type-2 fuzzy sets was introduced by Zadeh [15] as an extension of the concept of an ordinary fuzzy set, i.e., a type-1 fuzzy set. Type-2 fuzzy sets have grades of membership that are themselves fuzzy [17]. A type-2 membership grade can be any subset in $[0,1]$ – the *primary membership*; and, corresponding to each primary membership, there is a *secondary membership* (which can also be in $[0,1]$) that defines the possibilities for the primary membership. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity. Type-2 fuzzy sets allow us to handle linguistic uncertainties, as typified by the adage "words can mean different things to different people." A fuzzy relation of higher type (e.g., type-2) has been regarded as one way to increase the fuzziness of a relation, and, according to Hisdal, "increased fuzziness in a description means increased ability to handle inexact information in a logically correct manner [18]".

Figure 3 shows an example of a type-2 set. The domain of the membership grade corresponding to $x = 4$ is also shown. The membership grade for every point is a Gaussian type-1 set contained in $[0,1]$, we call such a set a "Gaussian type-2 set". When the membership grade for every point is a crisp set, the domain of which is an interval contained in $[0,1]$, we call such type-2 sets "interval type-2 sets" and their membership grades "interval type-1 sets". Interval type-2 sets are very useful when we have no other knowledge about secondary memberships.

3.2 Introduction to Type-2 Fuzzy Logic Systems: An Overview

Figure 4 shows the structure of a type-2 FLS. It is very similar to the structure of a type-1 FLS [16]. For a type-1 FLS, the *output processing* block only contains the defuzzifier. We assume that the reader is familiar with type-1 FLSs, so that here we focus only on the similarities and differences between the two FLSs.

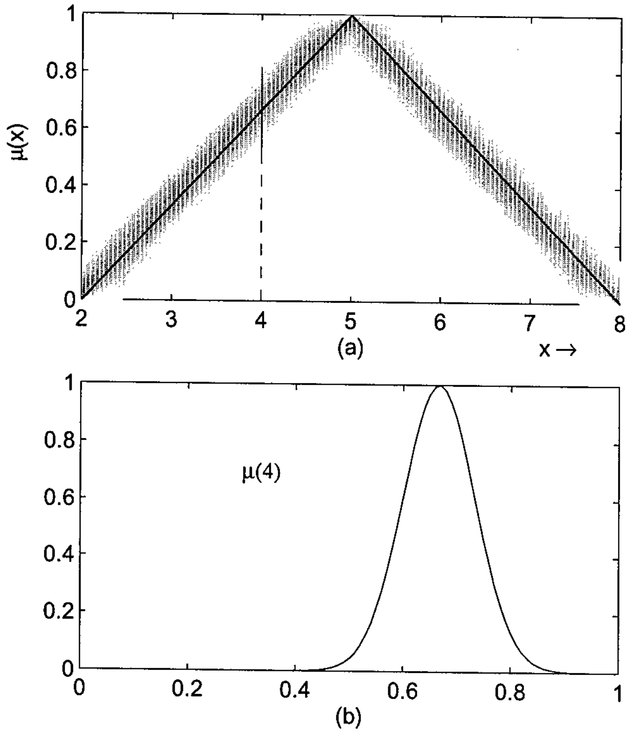


Figure 3: (a) Pictorial representation of a Gaussian type-2 set. The secondary memberships in this type-1 fuzzy set are shown in (b), and are Gaussian. Note that this set is called a Gaussian type-2 set because all its secondary membership functions are Gaussian. The "principal" membership function (the bold line), which is triangular in this case, can be of any shape.

The fuzzifier maps the crisp input into a fuzzy set. This fuzzy set can, in general, be a type-2 set.

In the type-1 case, we generally have "IF-THEN" rules, where the lth rule has the form "R^l : IF x_1 is F_1^l and x_2 is F_2^l and $\cdots$ and x_p is F_p^l, THEN y is G^l", where: x_is are inputs; F_i^ls are antecedent sets ($i = 1, \dots, p$); y is the output; and G^ls are consequent sets. The distinction between type-1 and type-2 is associated with the nature of the membership functions, which is not important while forming rules; hence, the structure of the rules remains exactly the same in the type-2 case, the only difference being that now some or all of the sets involved are of type-2; so, the lth rule in a type-2 FLS has the form "R^l : IF x_1 is $\tilde{\mathrm{F}}_1^l$ and x_2 is $\tilde{\mathrm{F}}_2^l$ and $\cdots$ and x_p is $\tilde{\mathrm{F}}_p^l$, THEN y is $\tilde{\mathrm{G}}^l$".

In the type-2 case, the inference process is very similar to that in type-1. The inference engine

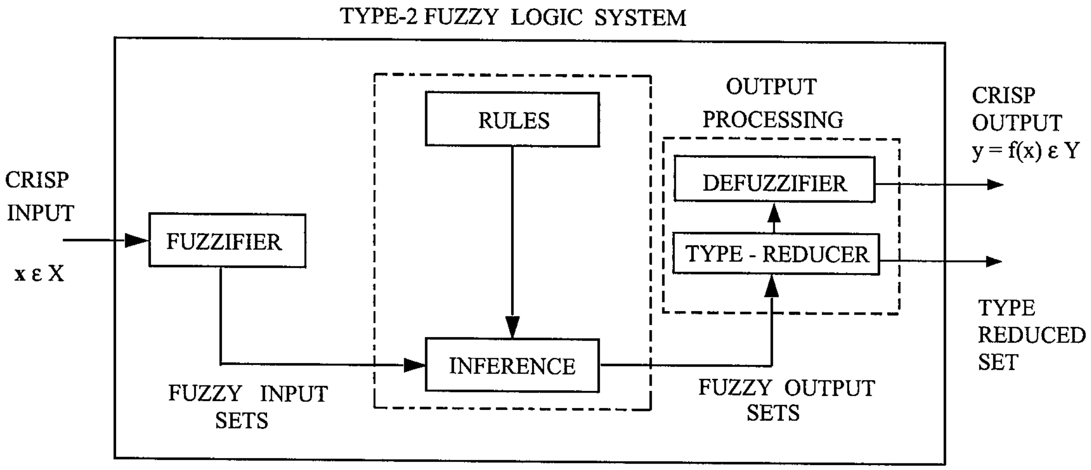


Figure 4: The structure of a type-2 FLS. In order to emphasize the importance of the type-reduced set, we have shown two outputs for the type-2 FLS, the type-reduced set and the crisp defuzzified value.

combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. To do this, one needs to find unions and intersections of type-2 sets, as well as compositions of type-2 relations.

In a type-1 FLS, the defuzzifier produces a crisp output from the fuzzy set that is the output of the inference engine, i.e., a type-0 (crisp) output is obtained from a type-1 set. In the type-2 case, the output of the inference engine is a type-2 set; so, we use "extended versions" (using Zadeh's Extension Principle [15]) of type-1 defuzzification methods. This extended defuzzification gives a type-1 fuzzy set. Since this operation takes us from the type-2 output sets of the FLS to a type-1 set, we call this operation "type-reduction" and the type-reduced set so obtained a "type-reduced set".

To obtain a crisp output from a type-2 FLS, we can defuzzify the type-reduced set. The most natural way of doing this seems to be by finding the centroid of the type-reduced set; however, there exist other possibilities like choosing the highest membership point in the type-reduced set.

General type-2 FLSs are computationally intensive, because type-reduction is very intensive. Things simplify a lot when secondary membership functions (MFs) are interval sets (in this case, the secondary memberships are either 0 or 1). When the secondary MFs are interval sets, we call the type-2 FLSs "interval type-2 FLSs". In [21], we proposed the theory and design of interval type-2 fuzzy logic systems (FLSs). We proposed an efficient and simplified method to compute the input and antecedent operations for interval type-2 FLSs, one that is based on a general inference formula for them. We introduced the concept of upper and lower membership functions (MFs) and illustrate our efficient inference method for the case of Gaussian primary MFs. We also proposed a method for designing an interval type-2 FLS in which we tune its parameters.

3.3 Applications of Interval Type-2 Fuzzy Logic Systems

We have developed theory and design methods for the most useful kind of type-2 fuzzy logic system (FLSs), interval type-2 FLSs [21], and have applied them to a number of very interesting applications, such as

1. Fading channel equalization [22] and co-channel interference elimination [23]. The channel states in a fading channel or channel with co-channel interferences are uncertain, and we validated that an interval type-2 fuzzy set, Gaussian primary membership function with uncertain mean, can be used to represent such uncertainties.

2. Network video traffic modeling and classification [19]. MPEG variable bit rate (VBR) traffic are very bursty. We validated that the I, P, and B frame sizes are log-normal with fixed mean and uncertain variance, so an interval type-2 fuzzy sets can be used to model the bursty video traffic and an interval type-2 fuzzy logic system with such type-2 fuzzy set are demonstrated performing much better than a Bayesian classifier.

3. Connection admission control for ATM network [20]. Connection admission control is actually a decision making problem. Different factors such as incoming real-time video/audio packet sizes, non-real time packet sizes, the buffer sizes are uncertain. We applied an interval type-2 fuzzy logic to handle these uncertainties, and achieved very good performance.

4 Modeling Node Lifetime with Gaussian Membership Functions

Wireless sensor nodes are severely energy constrained due to their compact form. To increase the lifetime of sensor networks, hardware design and protocol approaches for different layers must take energy efficiency into account. However, a foundamental question - "what is the nature of sensor network lifetime?" has not been answered yet. Since the lifetimes of individual nodes are not constants but random variables, it follows that the network lifetime is also a random variable. Recent research by Jain and Liang [7] discovered that in a wireless sensor network where the workloads are well-balanced, a single node lifetime behaves the nature of normal distribution as demonstrated in Fig 5.

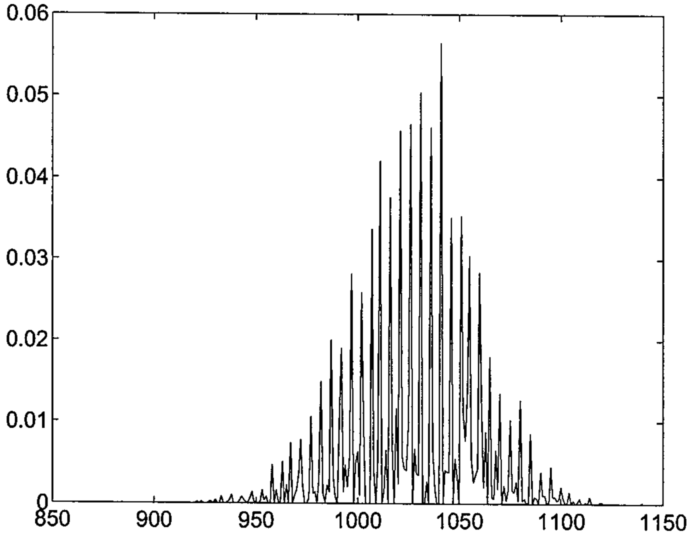


Figure 5: Single Node Lifetime Distribution

In this paper, we are first interested in setting up fine membership functions (MFs) for single node lifetime. From the original data of single node lifetime shown in Table 1, we decompose the whole data sets into seven segments, and compute the mean μ_i and standard deviation σ_i of node lifetime for each segment, $i = 1, 2, ...7$. The mean μ and standard deviation σ for of the entire data set is also computed. We are also interested to know which value - mean μ_i or standard deviation σ_i varies more. We first normalize the mean μ_i and standard deviation σ_i of each segment using

μ_i/μ and σ_i/σ. Then we compute the standard deviation of their normalized values σ_m and σ_{std}. Results are presented in Table 1.

From the last row of Table 1, we see that $\sigma_m << \sigma_{std}$ which means standard deviation σ_i varies much more than the mean value μ_i. Therefore we conclude that if the single node lifetime follows normal distribution, it is most appropriate to be modeled as a Gaussian MF with uncertain standard deviation. One example of type-2 Gaussian MF with uncertain standard deviation is shown in Fig 6. This result also justifies the use of the Gaussian MFs to model network lifetime in section 5.

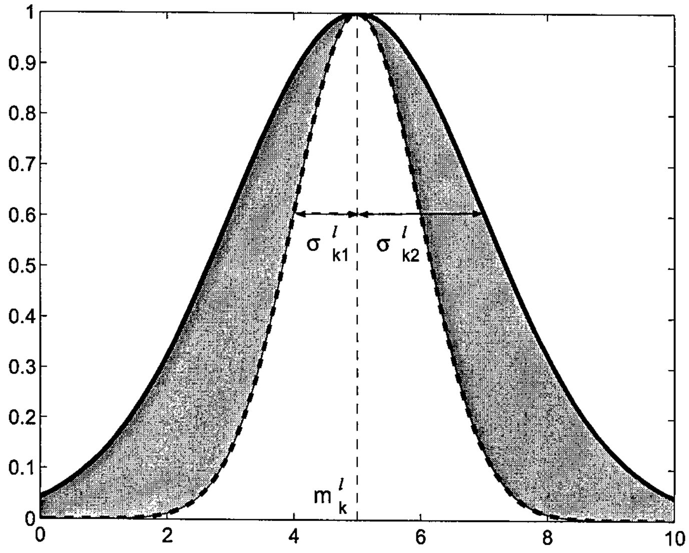


Figure 6: Type-2 Gaussian MF with uncertain standard deviation. The thick solid lines denote upper MFs, and the thick dashed lines denote lower MFs. The shaded regions are the footprints of uncertainty for interval secondaries. The center of the Gaussian MFs is 5, and the variance varies from 1.0 to 2.0

Table 1: MEAN AND STD VALUES FOR SEVEN SEGMENTS AND THE ENTIRE NODE LIFETIME, AND THEIR NORMALIZED STD

Node Lifetime Data	Mean	Std
Segment 1	1027.4	30.182
Segment 2	1028.9	29.819
Segment 3	1026.3	30.798
Segment 4	1028.7	30.917
Segment 5	1028	29.944
Segment 6	1027	29.975
Segment 7	1027.9	30.306
Entire Data Set	1027.7	30.292
Normalized STD	0.00082783	0.013105

5 Network Lifetime Analysis and Estimation Using Interval Type-2 Fuzzy Logic Systems

We are now ready to evaluate the network lifetime using interval type-2 fuzzy logic systems (FLSs). We apply reliability theory to design fuzzy rules. In the following section, we first treat the basics of reliability theory before extracting the knowledges for fuzzy rules. An interval type-2 FLS is then set up to perform lifetime analysis and estimation.

5.1 Fuzzy Rules Design Using Reliability Theory

Rules are the heart of a FLS, and may be provided by experts or can be extracted from numerical data. In either case, the rules that we are interested in can be expressed as a collection of IF-THEN statements, e.g.,

IF the noise of the wireless channel is high, THEN the quality of the received signal is poor.

For the lifetime issue studied in this paper, reliability theory provides a feasible method to design fuzzy rules. To understand this, we introduce the reliability block diagram (*RBD*). *RBD* is a graphical representation of the components of the system, and provides a visual representation of the way components are reliability-wise connected. Thus the effect of the success or failure of a component on the system performance can be evaluated.

Consider a system with two components. If this system is such that a single component failure can render the system nonfunctional, then we say that the components are reliability-wise, connected in series. If the system fails only when both its components fail, then we say that the components are reliability-wise connected in parallel. Note that the physical connection between the component may or may not be different from their reliability-wise connection. The *RBD*'s for both cases are given in Figs.7. Any complex system can be realized in the form of a combination of blocks connected in series and parallel.

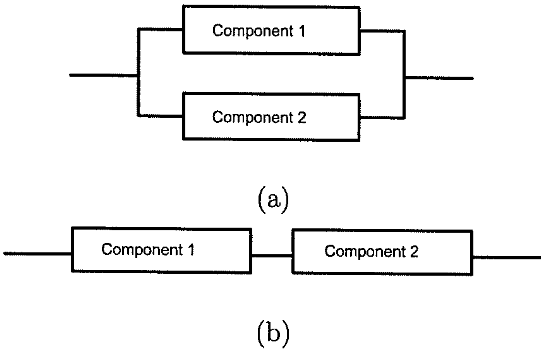


Figure 7: Reliability block diagrams (*RBD*) for a system of two components. (a)RBD with series connected components. (b)RBD with parallel connected components.

In our analysis, the wireless sensor network is the system under consideration and the sensor nodes are the components of the system. We demonstrate bellow how the knowledge is extracted for fuzzy rules design referring to the two examples in Fig.7.

Example 1: Set up Fuzzy Rules for Parallel System in Fig. 7(a)

In the parallel system, the system (network) fails only when both components (sensor nodes) fail. The rules can be set up as one example shown bellow:

IF the remaining battery level of component 1 (sensor node 1) is *low* and the remaining battery level of component 2 (sensor node 2) is *moderate*, THEN the lifetime of the system (network) is *moderate.*

Example 2: Set up Fuzzy Rules for Series system in Fig. 7(b)

In the series system, the system (network) fails when either component fails. The rules can be set up as one example shown bellow:

IF the remaining battery level of component 1 (sensor node 1) is *low* and the remaining battery level of component 2 (sensor node 2) is *moderate*, THEN the lifetime of the system (network) is *low.*

Note that the parallel and series systems are the two basic ways to model two sensor nodes. A wireless sensor network consisting of multiple sensor nodes can be first represented in the reliability block diagram. Fuzzy rules are then set up based on the above examples.

5.2 Simulation and Discussion

In our simulations, interval type-2 FLSs are constructed separately for square-grid netowrk and hex-grid network. We take the remaining energy level of each individual sensor node as input to the interval type-2 FLS and the output is the estimated network lifetime. As we have discussed in section 4, we choose type-2 Gaussian with uncertain standard deviation as the membership functions for both antecedent and consequent. The linguistic variables to represent the antecedent - remaining energy level are divided into three levels: *high*, *moderate* and *low* and the consequent - estimated network life is divided into five levels: *very high*, *high*, *moderate*, *low* and *very low.*

Simulations are performed for both squre-grid and hex-grid sensor networks. In both cases, 36 nodes are deployed and the distance between neighboring nodes is assumed to be the same. All sensor nodes are initialized with maximum energy level of 10 Joules. We assume the workload of the entire network is well-balanced, therefore energy dissipation of the single sensor node can be figured as linear most of the alive time. Our simulations are based on $N = 600$ lifetime data from the above well-balanced sensor network. The first 300 data are for training and the remaining 300 data are for testing. After training, the rules are fixed and we test the interval type-2 by evaluating the root mean square errors (RMSE) between the defuzzified output of FLS and the real lifetime data. We compare our interval type-2 FLSs with singleton type-1 FLSs. For both FLS schemes, we run 200 Monte-Carlo realizations and for each realization, each FLS is tuned using a simple steepest-descent algorithm for six epochs. Simulation results are averaged over all 200 Monte-Carlo realizations.

5.2.1 Square Grid

As defined in Section 2.4, the minimum network lifetime is the time to failure of any two neighboring nodes. We know that the failure of any single node does not cause network failure. The failure of any node coupled with the failure of any of its neighbors causes network failure. Using this definition we build the RBD for the square-grid as shown in Fig 8.

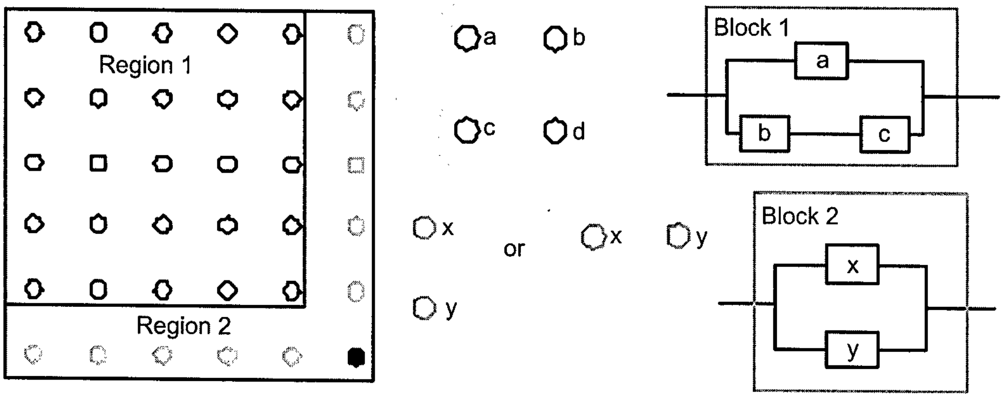


Figure 8: RBD of a single node in a square grid. Nodes belonging to region-1 are modeled as block-1 and nodes belonging to region-2 are modeled as block-2. The network RBD consists of $(\sqrt{N_{min}} - 1)^2$ block-1's in series with $2(\sqrt{N_{min}} - 1)$ block-2's

Fig 8 shows the *RBD* block for a single node in the network. A node can be modeled in two ways depending on its position in the sensor field. This distinction based on its position is made due to a simple observation that nodes at the right edge of the sensor field (region-2) do not have any right neighbor (node b) as opposed to nodes in region-1. Also, nodes at the bottom edge of the sensor field (region-2) do not have a bottom neighbor (node c) as opposed to the nodes in region-1. Note that as every node in a square-grid, node a has four neighbors, but its relationship with only two neighbors is modeled in its *RBD* block. This is because the relationship with the other two neighbors will be modeled when their *RBD* blocks are constructed. If this is not followed then the relationship between every node-neighbor pair will be modeled twice.

In this square grid network shown in Fig 8, we classify three antecedents based on the *RBD* of block-1:

- The remaining battery level of node a.
- The minimum remaining battery level of node b and c.
- The remaining battery level of node d.

We set up 27 rules for this FLS because every antecedent has 3 fuzzy sub-sets and there are 3 antecedents.

Two antecedents are chosen based on the *RBD* of block-2 in Fig 8 and total 9 rules are constructed in this case.

- The remaining battery level of node x.
- The remaining battery level of node y.

Let N_{min} be the number of sensor nodes required to be deployed with minimum density. The network *RBD* consists of $(\sqrt{N_{min}} - 1)^2$ block-1's and $2(\sqrt{N_{min}} - 1)$ block-2's in series, the whole square grid network can actually be decomposed into multiple blocks serial connected together and the method of setting up rules can be applied as well. Simulation results of RMSE is shown in Fig. 9.

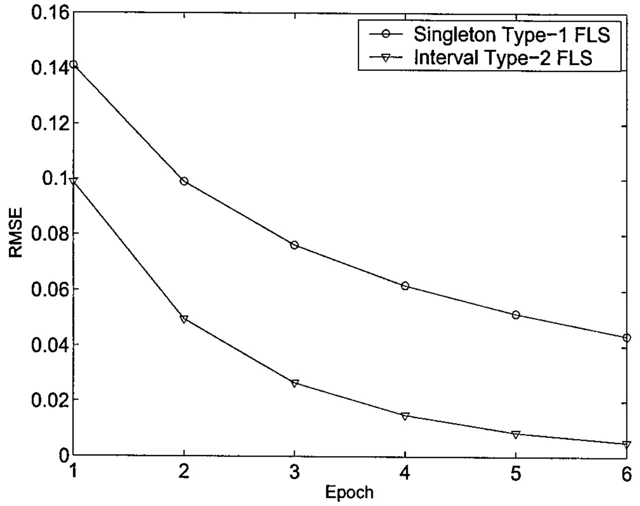


Figure 9: Square-Grid: The RMSE(for the test data) for two FLS approaches averaged over 200 Monte-Carlo realization

5.2.2 Hex-Grid

The analysis for the hex-grid is carried out on the same lines as that of the square-grid. Fig 2 (b) shows that as in the case of a square grid, two neighboring node failures cause network failure. The RBD block of a single node is shown in Fig. 10.

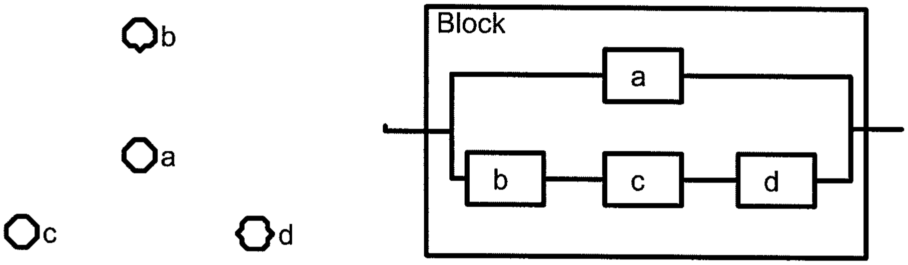


Figure 10: RBD block for a single node in the Hex-grid: The network RBD consists of $N/2$ such blocks in series.

Since the relation between a node and all of its neighbors is modeled by its corresponding *RBD* block, the *RBD* block's for the neighbors is not constructed as this causes the relationship between the nodes to be considered twice. In this case, we classify two antecedents and construct 9 rules.

- The remaining battery level of node a.
- The minimum remaining battery level of node b, c and d.

Since $N_{min}/2$ such blocks connected in series represent the network, the whole hex grid network can be decomposed the same way as in square grid networks. Simulation results of RMSE is shown in Fig. 11.

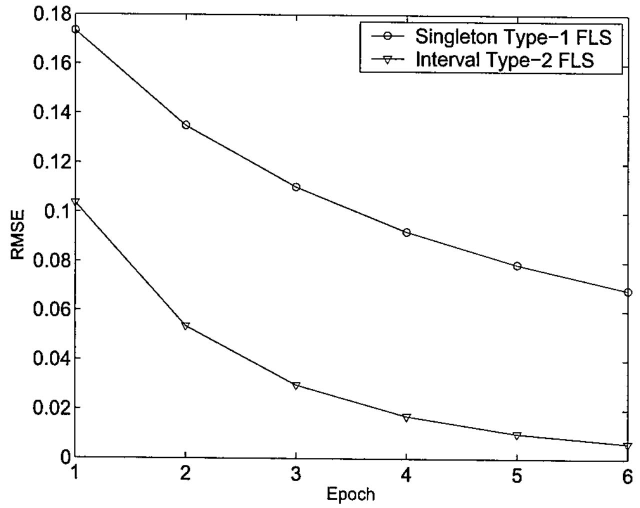


Figure 11: Hex-Grid: The RMSE(for the test data) for two FLS approaches averaged over 200 Monte-Carlo realization

Observe Fig.9 and Fig.11, interval type-2 FLSs outperform singleton type-1 FLSs. This results shows that the interval type-2 FLSs are more feasible for real-time energy estimation.

6 Conclusion and future works

In this paper, we describe a new method based on fuzzy logic theory to analyze and estimate the network lifetimes for wireless sensor networks. Our approach is illuminated by the discovery that a single node lifetime behaves the nature of normal distribution. However,we deem that if the single node lifetime follows normal distribution, it is most appropriate to be modeled as a Gaussian MF with uncertain standard deviation. We then set up the interval type-2 FLSs for energy estimation and evaluate their performance using real lifetime data. Simulation results justified the feasibility of applying type-2 FLSs into wireless sensor network lifetime analysis. Interval type-2 FLSs provides a way to handel knowledge uncertainty. We believe that our approaches opens up a new vision for research on sensor network lifetimes.

Our future work will focus on lifetime evaluation under the circumstances that the task scheduling is variable and how the estimated network lifetime could be used to accomodate the scheduling change.

Acknowledgment

This work was supported by the U.S. Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466, "Energy Efficient Wireless Sensor Networs for Future Combat System Using Fuzzy Logic".

References

[1] C.Y. Chong, S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges" *Proc. IEEE* , vol 91, no. 8 , Aug 2003, pp. 1247 -1256

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, "Wireless Sensor Networks for Habitat Monitoring" *Proc. WSNA'02* Atlanta, Georgia, Sep 28, 2002.

[3] V. A. Kottapalli, A. S. Kiremidjian, J. P. Lynch, Ed Carryer, T. W. Kenny, "Two-tired wireless sensor network architecture for structural health monitoring" *Proc. SPIE* San Diego, CA, Mar 2003.

[4] M. Bhardwaj, T. Garnett, A. Chandrakasan, "Upper Bounds on the Lifetime of Sensor Networks" *Proc. IEEE International Conference on Communications,* pp.785-790, 2001.

[5] M. Bhardwaj, A. P. Chandrakasan, "Bounding the Lifetime of Sensor Networks via Optimal Role Assignments" *Proc. INFOCOM 2002* pp:1587 - 1596, vol.3

[6] D. M. Blough, P. Santi, "Investigating Upper Bounds on Network Lifetime Extension for Cell-Based Energy Conservation Techniques in Stationary Ad Hod Networks" *Proc. MOBI-COM'2002* Atlanta, Georgia, Sep 2002

[7] E. Jain and Q. Liang, "Sensor placement and lifetime of wireless sensor networks: theory and performance analysis," *Sensor Network Operations*, edited by S. Phoha, T. F. La Porta, and C. Griffin, IEEE Press, Piscataway, NJ, 2004.

[8] N. N. Karnik and J. M. Mendel, *An Introduction to Type-2 Fuzzy Logic Systems*, October 1998, USC Report, http://sipi.usc.edu/~mendel/report.

[9] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems", *IEEE Trans. Fuzzy Systems*, vol. 7, no. 6, pp. 643-658, Dec. 1999.

[10] N. N. Karnik and J. M. Mendel, "Type-2 Fuzzy Logic Systems : Type-Reduction", presented at the *1998 IEEE SMC Conference*, San Diego, CA, October.

[11] N. N. Karnik, J. M. Mendel, and Q. Liang, "Centroid of a type-2 fuzzy set," *Information Sciences.* vol.132, pp.195-220, Feb, 2001,

[12] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, May/June 1993.

[13] K. Kar, S. Banerjee, "Node Placement for Connected Coverage in Sensor Networks" *Extended Abstract. Proc. WiOpt 2003* Sophia-Antipolis, France, March 2003.

[14] S. S. Dhillon, K. Chakrabarty, S. S. Iyengar, "Sensor Placement for Grid Coverage under Imprecise Detections," FUSION, 2002.

[15] L. A. Zadeh, " The concept of a linguistic variable and its application to approximate reasoning - I," *Information Sciences*, vol. 8, pp. 199-249, 1975.

[16] J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proc. of the IEEE*, vol. 83, no. 3, pp. 345-377, March 1995.

[17] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, USA, 1980.

[18] E. Hisdal, "The IF THEN ELSE statement and interval-valued fuzzy sets of higher type," *Int'l. J. Man-Machine Studies*, vol. 15, pp. 385-455, 1981.

[19] Q. Liang and J. M. Mendel, "MPEG VBR video traffic modeling and classification using fuzzy techniques," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 1, pp.183-193, Feb 2001.

[20] Q. Liang, N. Karnik, and J. M. Mendel, " Connection admission control in ATM network using survey-based type-2 fuzzy logic systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 30, no. 3, pp. 529-539, August 2000.

[21] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535-550, Oct 2000.

[22] Q. Liang and J. M. Mendel, "Equalization of time-varying nonlinear channels using type-2 fuzzy adaptive filters," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 5, pp. 551-563, Oct 2000.

[23] Q. Liang and J. M. Mendel, " Overcoming time-varying co-channel interference using type-2 fuzzy adaptive filters", *IEEE Transactions on Circuits and Systems, II*, vol. 47, no. 12, pp. 1419-1428, Dec 2000.

[24] L.-X. Wang and J. M. Mendel, " Fuzzy basis functions, universal approximation, and orthogonal least squares learning, " *IEEE Trans. on Neural Networks*, vol. 3, pp. 807-814, Sept. 1992.

[25] D. Kececioglu, "Reliability Engineering Handbook" *Volume 1 and 2,* Prentice Hall, Ney Jersey 1991.

[26] L. M. Leemis, "Reliability: Probabilistic Models and Statistical Methods," Prentice- Hall, 1995

[27] Life Data Analysis Reference. [Online] Available: http://www.weibull.com/lifedatawebcontents.htm

[28] A. Papoulis, S. U. Pillai "Probability, Random Varibales and Stochastic Processes, " *4th ed,* McGraw-Hill, New York 2002.

[29] B. Healy, "The Use of Wireless Sensor Networks for Mapping Environmental Conditions in Buildings" *ASHRAE Seminar, July 2 2003* Available Online: http://www.nist.gov/tc75/ASHRAESummer2003SeminarHealy.pdf

Fuzzy Optimization for Distributed Sensor Deployment

Haining Shu, Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016 USA
E-mail: shu@wcn.uta.edu, liang@uta.edu

Abstract— **The effectiveness of distributed wireless sensor networks highly depends on the deployment of sensors. Given a finite number of sensors, optimizing the sensor deployment will enhance the field coverage of a wireless sensor network. Network lifetime and quality of communication in terms of outage probability as a result, will be greatly ameliorated as the topology approaches uniformity fast. In this paper, we propose a fuzzy optimization algorithm (FOA) to efficiently adjust the sensor placement after an initial random deployment. We apply fuzzy logic theory to handle the uncertainty in sensor deployment problem. Simulation results show that our approach achieves fast and stable deployment and maximize the field coverage. Outage probabiliy, as a measure of communication quality gets effectively decreased.**

I. Introduction

Sensor networks consist of certain amount of small and energy constrained nodes. Sensor nodes are deployed in support of various missions including environment monitoring, battlefield surveillance, and emergency search and rescue.

A number of applications require the placement of sensors at desired locations. Such placement-friendly sensor networks are widely used for infrastructure security [1], where critical buildings and facilities such as airports and power plants are monitored by a network of sensors placed adequately.

Meanwhile, other applications employ sensor nodes with certain mobility like mobile robots. Mobile sensors are practically desirable for they have the capability to move around and re-adjust their positions for high quality communication and better surveillance [2].

Our primary interest lies in the wireless sensor network comprised of mobile sensors. Our goal is to optimize the sensor deployment such that the maximum field coverage and high quality communication could be achieved.

Some prior research proposed a strategy based on virtual forces in sensor deployment and target localization [3]. A distributed self-spreading algorithm was introduced in [4] to improve the network coverage. Poduri et al [5] proposed an algorithm with the constraint that each of the nodes has at lease K neighbors. These algorithms have made lots efforts to formulate the virtual forces, however none of which can well handle the random move and unpredictable oscillation in deployment.

Our algorithm is very different from all previous works. Instead of attempting to formulate the virtual forces, we propose to use fuzzy logic system in control of the sensor movement. To save battery life, we refer our appoach to open-loop control systems, in which the control action is independent of the physical system output, e.g. feedback control. By applying this fuzzy optimization mechanism to each individual mobile sensor, uncertain exhaustive move and oscillation is efficiently avoided and fast deployment is achieved. The entire network as a result, survives for longer lifetime and the quality of communication in terms of outage probability is greatly ameliorated as the topology approaches uniformity. A concept of coherence time is introduced for the purpose of synchronization among sensors.

This paper is organized as follows. In section II, we briefly review the basic concept of fuzzy logic systems. Section III details the Fuzzy Optimization Algorithm (FOA) designed for sensor deployment. Simulation and the key results of this work are presented in Section IV. Section V concludes with a summary. Fenton-Wilkinson method to tackle the outage problem is expatiated in appendix.

II. Overview of Fuzzy Logic Systems

Figure 1 shows the structure of a fuzzy logic system (FLS) [6]. When an input is applied to a FLS, the inference engine computes the output set corresponding to each rule. The defuzzifer then computes a crisp output from these rule output sets. Consider a p-input 1-output FLS, using singleton fuzzification, *center-of-sets* defuzzification [7] and "IF-THEN" rules of the form [8]

$$R^l : \text{IF } x_1 \text{ is } \mathrm{F}_1^l \text{ and } x_2 \text{ is } \mathrm{F}_2^l \text{ and } \cdots \text{ and } x_p \text{ is } \mathrm{F}_p^l, \text{ THEN } y \text{ is } \mathrm{G}^l.$$

Assuming singleton fuzzification, when an input $\mathbf{x}' = \{x_1', \ldots, x_p'\}$ is applied, the degree of firing corresponding to the lth rule is computed as

$$\mu_{\mathrm{F}_1^l}(x_1') \star \mu_{\mathrm{F}_2^l}(x_2') \star \cdots \star \mu_{\mathrm{F}_p^l}(x_p') = \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i') \quad (1)$$

where $\star$ and $\mathcal{T}$ both indicate the chosen t-norm. There are many kinds of defuzzifiers. In this paper, we focus, for illustrative purposes, on the center-of-sets defuzzifier [7]. It computes a crisp output for the FLS by first computing the centroid, c_{G^l}, of every consequent set G^l, and, then computing a weighted average of these centroids. The weight corresponding to the lth rule consequent centroid is the degree of firing associated with the lth rule, $\mathcal{T}_{i=1}^{p}\mu_{F_i^l}(x_i')$, so that

$$y_{cos}(\mathbf{x}') = \frac{\sum_{l=1}^{M} c_{G^l}\mathcal{T}_{i=1}^{p}\mu_{F_i^l}(x_i')}{\sum_{l=1}^{M} \mathcal{T}_{i=1}^{p}\mu_{F_i^l}(x_i')} \tag{2}$$

where M is the number of rules in the FLS.

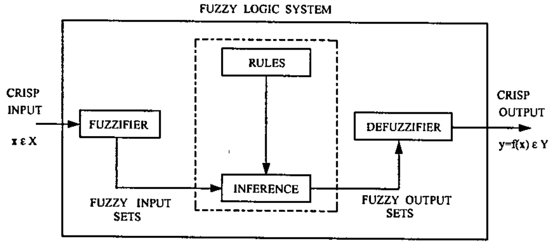


Fig. 1. The structure of a fuzzy logic system.

III. Extracting the Knowledge for Fuzzy Optimization Algorithm

A. Assumptions and Notations

- Sensor field is denoted by a two-dimensional grid.
- Coverage discussed in this paper is grid coverage.
- A grid point is covered when at least one sensor covers this point.
- A sensor can detect or sense any event within its sensing range, denoted by Rs. Coverage is determined based on Rs.
- Two sensors within their communication range, denoted by Rc can communicate with each other.
- Neighbors of a sensor are nodes within its communication range.
- Detection and communication is modeled as a circle on the two-dimensional grid.

B. Fuzzy Optimization Algorithm

The Fuzzy Optimization Algorithm (FOA) is illuminated by the powerful capability of fuzzy logic system to handle uncertainty and ambiguity. Fuzzy logic system is well known as model free. Their membership functions are not based on statistical distributions. In this paper, we apply fuzzy logic system to re-position the sensor nodes and optimize the network deployment.

Our algorithm starts with random deployment. Initially, a given number of sensors are randomly deployed in a square sensor field. We have made the following assumptions:

- All sensor nodes are peer to peer.
- Sensor nodes have certain mobility and capabilities of computing, detection and communication.
- Sensor node knows its location information.
- Sensors are synchronized by coherence time. One-time move is made within each coherence period.

Two critical procedures are considered in our algorithm:

- Determine the next-step move distance for each sensor.
- Determine the next-step move direction for each sensor.

The next-step move distance is hard to determine. Too small or big move distance each step consumes the network more time and energy to get stable deployment. Excessive oscillation is unavoidable in previous work with no fuzzy control. In this paper, we design a fuzzy logic system to determine the next-step move distance for each sensor.

We collect the knowledge for deployment problem based on the following two antecedents:

Antecedent 1. Number of neighbors of each sensor.

Antecedent 2. Average Euclidean distance between sensor node and its neighbors

The linguistic variables to represent the number of neighbors for each sensor are divided into three levels: *high*, *moderate* and *low*; and those to represent the average Euclidean distance between sensor node and its neighbors are divided into three levels: *far*, *moderate* and *near*. The consequent - the shift distance normalized by sensing range is divided into three levels: *far*, *moderate* and *near*. Table 1 summaries the rules and consequents.

TABLE I
Fuzzy Rules and Consequent

Antecedent1	*Antecedent2*	*Consequent*
Low	*Near*	*Moderate*
Low	*Moderate*	*Near*
Low	*Far*	*Near*
Moderate	*Near*	*Far*
Moderate	*Moderate*	*Moderate*
Moderate	*Far*	*Near*
High	*Near*	*Far*
High	*Moderate*	*Moderate*
High	*Far*	*Moderate*

One example of rules is as follows:

IF the number of neighbors of sensor *i* is *high* and average Euclidean distance between sensor *i* and its neighbors is *moderate*, THEN the normalized scalar shift distance of sensor *i* will be *high*.

We set up 9 rules for this FLS because every antecedent has 3 fuzzy sub-sets and there are 2 antecedents. Trapezoidal membership functions (MFs) are used to represent *high*, *low*, *far* and *near* and triangle MFs to represent *moderate*. We show these membership functions in Figure 2.

Applying center-of-sets defuzzification [7], for every input (x_1, x_2), the output is computed using

$$y_{(x_1,x_2)} = \frac{\sum_{l=1}^{9} c_{G^l}\mu_{F_1^l}(x_1)\mu_{F_2^l}(x_2)}{\sum_{l=1}^{9} \mu_{F_1^l}(x_1)\mu_{F_2^l}(x_2)} \tag{3}$$

Repeating these calculations for $\forall x_i \in [0,1]$,we obtain a control surface $y(x_1, x_2)$ as shown in Figure 3.

The concept of control surface, or decison surface, is central in fuzzy logic systems. It describes the dynamics of the

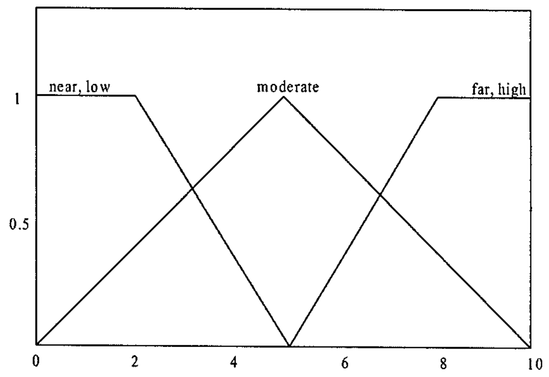


Fig. 2. Antecedent Membership Function

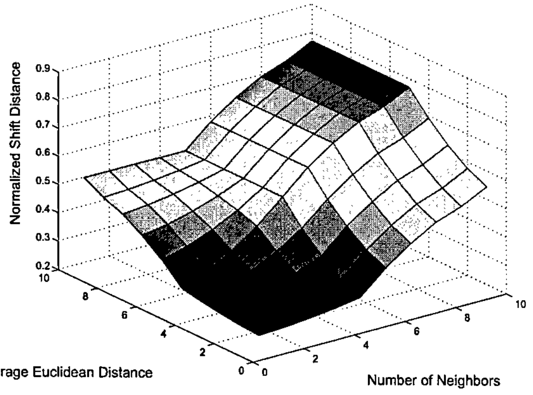


Fig. 3. Control Surface

controller and is generally a time-varying nonlinear surface. From figure 3, we can see that although the number of neighbors for a certain sensor is high, the move distance can be nearer than some sensor with fewer "crowded" neighbors, i.e. very close average Euclidean distance between the sensor and its neighbors. With the assist of control surface, the next-step move distance can be carefully determined.

Comparing to move distance, the next-step move direction is much easier to decide. Coulomb's law in physics becomes a useful tool to tackle the problem. Assume sensor i has 2 neighbors as shown in Figure 4.

The coordinate of sensor i is denoted as C_i = (X_i,Y_i).

The next-step move direction of sensor i could be represented as follows:

$$\vec{v}=\sum_{j=1}^{m}\frac{\vec{C_j}-\vec{C_i}}{|\vec{C_j}-\vec{C_i}|^2} \qquad (4)$$

$$tan(\alpha)=\frac{Y_{(\vec{v})}}{X_{(\vec{v})}} \qquad (5)$$

After getting distance and direction (angleα) , sensor i clearly knows his next-step move position. In order to prolong the battery life of each individual sensor, we introduce a coherence time as the duty cycle of executing the FOA algorithm. Sensors are put into idle or sleep mode if within the coherence time, the information of neighbors remains unchanged.

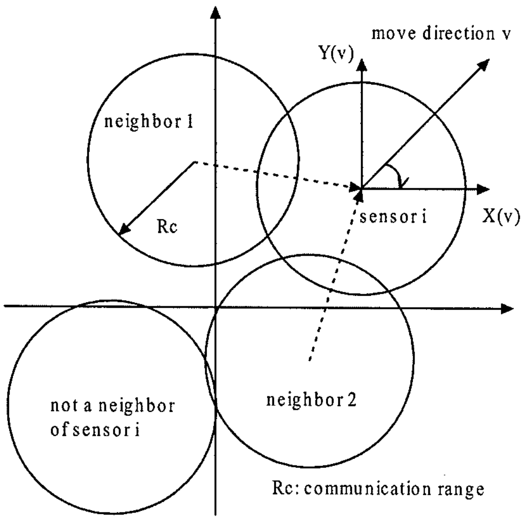


Fig. 4. Next Step Move Direction

IV. Simulation and Discussion

We investigate various number of sensors deployed in a field of 10×10 square kilometers area. We assume each sensor is equipped with an omni antenna to carry out the task of detection and communication. Evaluation of our FOA algorithm follows three criteria: field coverage, outage probability and convergence. Results are averaged over 200 Monte Carlo simulations.

Figure 5 shows at R_s=1km and R_c=2km, the coverage of the initial random deployment and the one after FOA algorithm. The FOA algorithm could improve the network coverage by 20% - 30% in average. As 60 sensors are deployed, the coverage approaches to 100% after FOA algorithm is implemented.

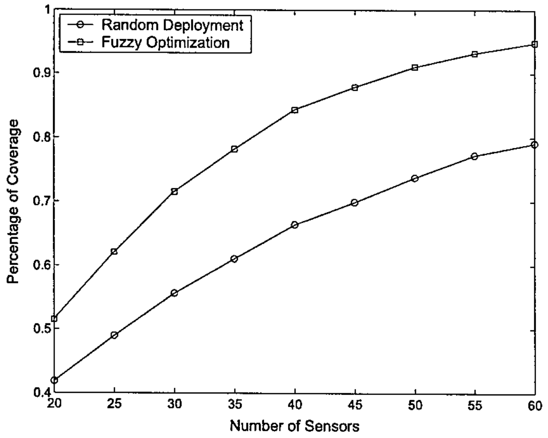


Fig. 5. Coverage vs. Number of Nodes (Rc=2,Rs=1)

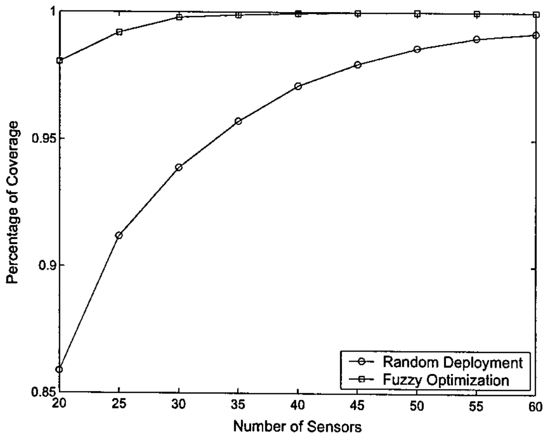


Fig. 6. Coverage vs. Number of Nodes (Rc=4,Rs=2)

Figure 6 gives the results when R_s=2km and R_c=4km, the coverage comparison between random deployment and fuzzy optimization algorithm. In the case when 20 sensors are deployed, initialy the coverage after random deployment is around 85%. After FOA algorithm is excuted, the coverage reaches 98%. The coverage is dramatically improved in the low density network. The above two figures indicate that instead of deploying large amount of sensors, the desired field coverage could also be achieved with fewer sensors.

In celluar radio systems the radio link performance is usually limited by interference rather than noise, therefore, the probability of outage due to co-channel interference is of primary concern. Measurements [9] have shown that at any value of $d_{i,j}$ (the Euclidean distance between sensor i and sensor j), the path loss $PL(d_{i,j})$ is random and distributed log-normally (normal in dB) about the mean distance dependent value. That is:

$$PL(d_{i,j})[dB] = \overline{PL}(d_{i,j}) + X_\sigma = \overline{PL}(d_0) + 10nlog(\frac{d_{i,j}}{d_0}) + X_\sigma \tag{6}$$

and

$$P_r(d_{i,j})[dBm] = P_t[dBm] - PL(d_{i,j})[dB] \tag{7}$$

where X_σ is a zero-mean Gaussian distribution random variable (in dB) with standard deviation σ (also in dB).

The log-normal distribution describes the random *shadowing* effects on the propagation path which implies that measured signal levels at certain distance have a Gaussian (normal) distribution about the distance-dependent mean and standard deviation σ. Since $PL(d_{i,j})$ follows normal distribution, so is $P_r(d_{i,j})$, and the Q function may be used to determine the probability that the received signal level will exceed (or fall below) a particular level.

The probability that the received signal level will exceed a certain value γ can be calculated from the cumulative density function as

$$P_r[P_r(d_{i,j}) > \gamma] = Q\left(\frac{\gamma - \overline{P_r}(d_{i,j})}{\sigma}\right) \tag{8}$$

For sensor i with N neighbors, if sensor i acts as the destination node during one communication, the signal to interference ratio (SIR) is represented as:

$$SIR(i) = \frac{P_r(d_{i,j})}{\sum_{k=1}^{N} P_r(d_{i,k})}, k \neq j \tag{9}$$

The denominator denoting the effect of co-channel interference is a sum of $N-1$ log-normal signals. Evaluating the outage probability requires the probability distribution of the interference power. There is no known exact expression for the probability distribution for the sum of log-normal random variables, but various authors have derived several approaches which approximate the sum of log-normal random variables by another log-normal random variable.

In this paper, we introduce Fenton-Wilkinson method [10]. The co-channel interference can now be approximated by one log-normal random variable. SIR(in dB) as a result follows log-normal distribution as well. We expatiate the Fenton-Wilkinson method in the appendix. Results of outage probability are presented in figure 7.

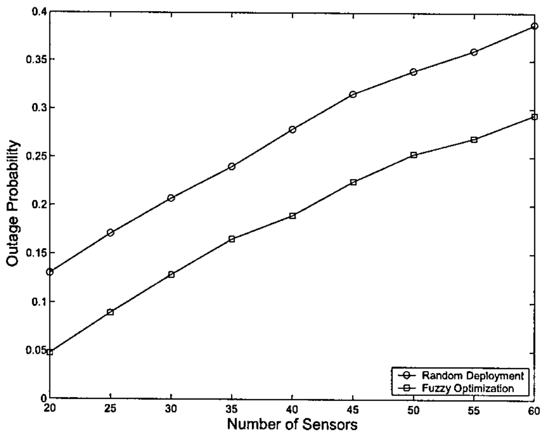


Fig. 7. Outage Probability vs. Number of Sensors (Rc=4,Rs=2)

From Figure 7, we can see that our FOA algorithm successfully reduced the outage probability by nearly 30% which implies a higher probability that the received signal level will exceed the SIR threshold. The quality of communications then can be greatly ameliorated.

The performance of FOA algorithm can also be evaluated in terms of convergence speed. We demonstrate the coverage at each iteration e.g. coverage at the ith iteration is $Cov(i)$ for different number of sensors and the mean square errors (MSE) between adjacent iterations shown as below.

$$MSE(i) = \frac{[Cov(i) - Cov(i-1)]^2}{Cov(i-1)^2} i = 1, ..., iteration \tag{10}$$

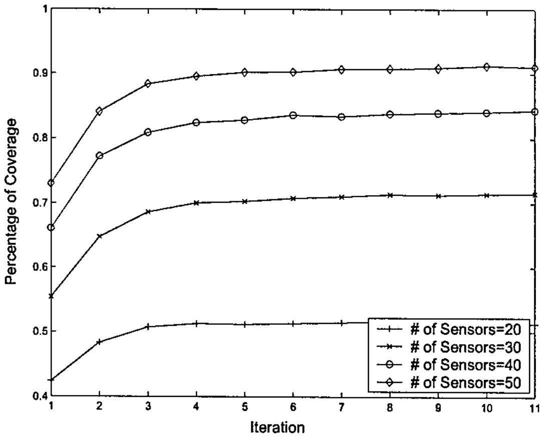


Fig. 8. Coverage vs.Interation (Rc=2,Rs=1)

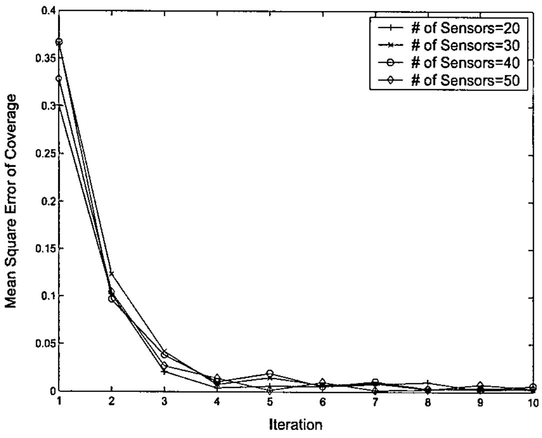


Fig. 10. MSE vs. Coverage (Rc=2,Rs=1)

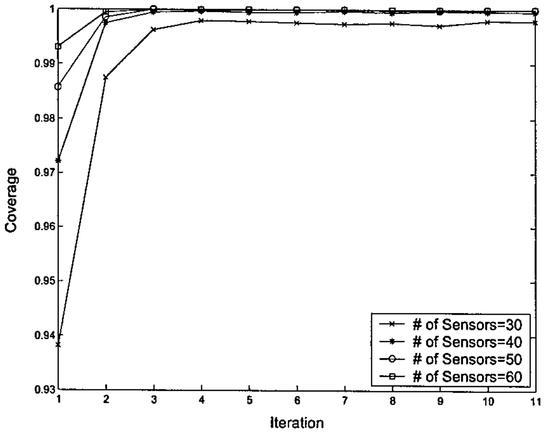


Fig. 9. Coverage vs.Interation (Rc=4,Rs=2)

Figure 8 and 9 show different number of sensors have approximate same convergence speed. The coverage increases vastly within the first 2 to 3 iterations. After that coverage approaches stable for all different number of sensors. Figure 9 shows at R_s=1km and R_c=2km the coverage mean square errors between the adjacent iterations. The results also validate that the convergence of FOA algorithm is independent of the number of sensors to be deployed.

Comparing to other algorithm such as distributed self spreading algorithm [4] which takes over 20 termination times and 10 oscillations and status limit to achieve the similar coverage improvement, our FOA algorithm is much simpler to implement and outperforms in a fast and guaranteed convergence.

V. Conclusions

In this paper, we proposed a new sensor deployment strategy - Fuzzy Optimization Algorithm based on fuzzy logic system. Our approach has a great advantage to deal with the randomness in sensor deployment which is particularly useful when emergency rescue or redeployment over hostile situation is needed. We believe that in an energy constraint wireless sensor network, fast and efficient deployment strategy is a necessity to save battery power and extend network lifetime. Our FOA algorithm is capable to model all random deployment with a fuzzy logic system. The network coverage as a result gets greatly improved and quality of communication in term of outage probablity is ameliorated. Moreover, the FOA algorithm brings the whole network to a stable and optimal deployment very soon which will significantly reduce the energy consumption. Our future work will focus on modeling the random deployment with some existing pattern so that the energy consumption can be further studied in the deployment problem.

Acknowledgment

This work was supported by the Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466.

References

[1] C.Y.Chong and S.P.Kumar "Sensor Networks: Evolution, Opportunities, and Challenges" *Proc.IEEE*, vol. 91, no. 8, pp. 1247-1256, Aug. 2003.

[2] H.Qi,S.S.Iyengar and K.Chakrabarty "Distributed Sensor fusion - a review of recent research" *Journal of the Franklin Institute*, vol. 338, pp. 655-668, 2001.

[3] Y.Zhou and K. Chakrabarty "Sensor deployment and target localization based on virtual forces" *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 2, pp. 1293-1303, 2003.

[4] N.Heo and P.K.Varshney "A distributed self spreading algorithm for mobile wireless sensor networks" *Wireless Communications and Networking, IEEE International Conference on*, vol. 3, pp. 1597 - 1602, 2003.

[5] S.Poduri and G.S.Sukhatme "Constrained coverage for mobile sensor networks" *Robotics and Automation, IEEE International Conference on*, vol. 1, pp. 165 - 171,2004

[6] J.M.Mendel "Fuzzy Logic Systems for Engeneering: A Tutorial" *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345 - 377,1995

[7] J.M.Mendel "Uncertain Rule-Based Fuzzy Logic Systems" *Prentice-Hall, Upper Saddle River, NJ*, 2001

[8] E. H. Mamdani, " Applications of fuzzy logic to approximate reasoning using linguistic systems", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 26, no. 12, pp. 1182-1191, 1977.

[9] T.S.Rappaport, "Wireless communications:principles and practice" *Prentic e-Hall, Upper Saddle River, NJ*, 2001

[10] G.L.Stuber "Principles of mobile communication" *Boston : Kluwer Academic Publishers,* 2001

Appendix

A. Multiple Log-Normal Interferers

Consider the sum of N_I log-normal random variables

$$I = \sum_{k=1}^{N_I} \Omega_k = \sum_{k=1}^{N_I} 10^{\Omega_{k(dBm)}/10} \tag{11}$$

where the $\Omega_{k(dBm)}$ are Gaussian random variables with mean $\mu_{\Omega_{k(dBm)}}$ and variance $\sigma^2_{\Omega_k}$, and the $\Omega_k = 10^{\Omega_{k(dBm)}/10}$ are the log-normal random variables. Unfortunately, there is no known closed form expression for the probability density function (pdf) of the sum of multiple ($N_I \geq 2$) log-normal random variables. However, there is a general consensus that the sum of independent log-normal random variables can be approximated by another log-normal random variable with appropriately chosen parameters. That is,

$$I = \sum_{k=1}^{N_I} 10^{\Omega_{k(dBm)}/10} \approx 10^{Z_{(dBm)}/10} = \hat{I} \tag{12}$$

where $Z_{(dBm)}$ is a Gaussian random variable with mean $\mu_{Z(dBm)}$ and variance σ^2_Z. The problem is to determine $\mu_{Z(dBm)}$ and variance σ^2_Z in terms of the $\mu_{\Omega_{k(dBm)}}$ and variance $\sigma^2_{\Omega_k}$, $k = 1, ..., N_I$. Several methods have been suggested in the literature to solove this problem including those by Fenton, Schwartz and Yen, and Farley. Each of these methods provides varying degrees of accuracy over specified ranges of the shadow standard deviation σ_Ω, the sum I, and the number of interferes N_I.

B. Fenton-Wilkinson Method

With the Fenton-Wilkinson method, the mean $\mu_{Z(dBm)}$ and variance σ^2_Z of $Z_{(dBm)}$ are obtained by matching the first two moments of the sum I with the first two moments of the approximation $\hat{I}$. To derive the appropriate moments, it is convenient to use natural logarithms. We write

$$\Omega_k = 10^{\Omega_{k(dBm)}/10} = e^{\epsilon \Omega_{k(dBm)}} = e^{\hat{\Omega}_k} \tag{13}$$

where $\epsilon = (ln10)/10 = 0.23026$ and $\hat{\Omega}_k = \epsilon \Omega_{k(dBm)}$. Note that $\mu_{\hat{\Omega}_k} = \epsilon \mu_{\Omega_{k(dBm)}}$ and $\sigma^2_{\hat{\Omega}_k} = \epsilon^2 \sigma^2_{\Omega_k}$. The nth moment of the log-normal random variable Ω_k can be obtained from the moment generating function of the Gaussian random variables $\hat{\Omega}_k$ as

$$E[\Omega^n_k] = E[e^{n\hat{\Omega}_k}] = e^{n\mu_{\hat{\Omega}_k} + (1/2)n^2\sigma^2_{\hat{\Omega}_k}} \tag{14}$$

To find the appropriate moments for the log-normal approximation we can use (14) and equate the first two moments on both sides of the equation

$$I = \sum_{k=1}^{N_I} e^{\hat{\Omega}_k} \approx e^{\hat{Z}} = \hat{I} \tag{15}$$

where $\hat{Z} = \epsilon Z_{(dBm)}$. For example, suppose that $\hat{\Omega}_k$, $k = 1, ..., N_I$ have mean $\mu_{\hat{\Omega}_k}$, $k = 1, ..., N_I$ and identical variances $\sigma^2_{\hat{\Omega}}$. Identical variances are often assumed because the standard deviation of log-normal shadowing is largely independent of the radio path length. Equating the means on both sides of (15)

$$\mu_I = E[I] = \sum_{k=1}^{N_I} E[e^{\hat{\Omega}_k}] = E[e^{\hat{Z}}] = E[\hat{I}] = \mu_{\hat{I}} \tag{16}$$

gives the result

$$\left(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}} \right) e^{(1/2)\sigma^2_{\hat{\Omega}}} = e^{\mu_{\hat{Z}} + (1/2)\sigma^2_{\hat{Z}}} \tag{17}$$

Likewise, we can equate the variances on both sides of (15) under the assumption that the $\hat{\Omega}_k$, $k = 1, ..., N_I$ are independent

$$\sigma^2_I = E[I^2] - \mu^2_I = E[\hat{I}^2] = \sigma^2_{\hat{I}} \tag{18}$$

giving the result

$$\left(\sum_{k=1}^{N_I} e^{2\mu_{\hat{\Omega}_k}} \right) e^{\sigma^2_{\hat{\Omega}}} (e^{\sigma^2_{\hat{\Omega}}} - 1) = e^{2\mu_{\hat{Z}}} e^{\sigma^2_{\hat{Z}}} (e^{\sigma^2_{\hat{Z}}} - 1) \tag{19}$$

By squaring each side of (17) and dividing each side of resulting equation by the respective side of (19) We can solve for σ^2_{hatz} in terms of the known values of $\mu_{\hat{\Omega}_k}$, $k = 1, ..., N_I$ and $\sigma^2_{\hat{\Omega}}$. Afterwards, $\mu_{\hat{Z}}$ can be obtained from (17). This procedure yields the following solution:

$$\mu_{\hat{Z}} = \frac{\sigma^2_{\hat{\Omega}} - \sigma^2_{\hat{Z}}}{2} + ln \left(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}} \right) \tag{20}$$

$$\sigma^2_{\hat{Z}} = ln \left((e^{\sigma^2_{\hat{\Omega}}} - 1) \frac{\sum_{k=1}^{N_I} e^{2\mu_{\hat{\Omega}_k}}}{(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}})^2} + 1 \right) \tag{21}$$

Finally, $\mu_{Z(dBm)} = \epsilon^{-1}\mu_{\hat{Z}}$ and $\sigma^2_Z = \epsilon^{-2}\sigma^2_{\hat{Z}}$.

The accuracy of this log-normal approximation can be measured in terms of how accurately the first two moments of $I_{(dB)} = 10log_{10}I$ are estimated, and how well the cumulative distribution function (cdf) of $I_{(dB)}$ is described by a Gaussian cdf. In problems relating to the co-channel interference outage in celluar radio systems, we are usually interested in the tails of both the complementary distribution function (cdfc) $F^C_I = P(I \geq x)$ and the cdf $F_I(x) = 1 - F^C_I = P(I < x)$. In this case, we are interested in the accuracy of the approximation

$$F_I(x) \approx P(e^{\hat{Z}} \geq x) = Q\left(\frac{lnx - \mu_{\hat{Z}}}{\sigma_{\hat{Z}}} \right) \tag{22}$$

for large and small values of x. It will be shown later that the Fenton-Wilkinson method can approximate the tails of the cdf and cdfc functions with good accuracy.

Distributed Sensor Deployment Using Fuzzy Logic Systems

Haining Shu and Qilian Liang

Department of Electrical Engineering

University of Texas at Arlington

Arlington, TX 76019-0016 USA

E-mail: shu@wcn.uta.edu, liang@uta.edu

Abstract

The effectiveness of distributed wireless sensor networks highly depends on the deployment of sensors. Given a finite number of sensors, optimizing the sensor deployment will enhance the field coverage of a wireless sensor network. Network lifetime and quality of service in terms of outage probability as a result, will be greatly ameliorated as the topology approaches uniformity fast. In this paper, we apply fuzzy logic systems to efficiently adjust the sensor placement after an initial random deployment. To evaluate the outage probability due to co-channel interference, Fenton-Wilkinson method is used to approximate the sum of log-normal random variables. Simulation results show that our approach achieves fast and stable deployment and maximizes the field coverage. Outage probabiliy, as a measure of communication quality gets effectively decreased.

1 Introduction

Sensor networks consist of certain amount of small and energy constrained nodes. Sensor nodes are deployed in support of various missions including environment monitoring, battlefield surveillance, and emergency search and rescue.

A number of applications require the placement of sensors at desired locations. Such placement-friendly sensor networks are widely used for infrastructure security [1], where critical buildings and facilities such as airports and power plants are monitored by a network of sensors placed adequately.

Meanwhile, other applications employ sensor nodes with certain mobility like mobile robots. Mobile sensors are practically desirable because they have the capability to move around and readjust their positions for high quality communication and better surveillance [2].

Our primary interest lies in the wireless sensor network comprised of mobile sensors. Our goal is to optimize the sensor deployment such that fast deployment, maximum field coverage and high quality communication could be achieved.

Some prior research proposed a strategy based on virtual forces in sensor deployment and target localization [3]. A distributed self-spreading algorithm was introduced in [4] to improve the network coverage. Poduri et al [5] proposed an algorithm with the constraint that each of the nodes has at lease K neighbors. These algorithms have made lots efforts to formulate the virtual forces, however none of which can well handle the uncertainties such as random move and unpredictable oscillation in deployment. Various constraints like oscillation limit, stable status and number of neighbors [5] therefore have to be imposed to avoid excessive sensor movement.

In this paper, we apply fuzzy logic to handle these uncertainties and design fuzzy logic systems (FLSs) for distributed sensor deployment. Instead of attempting to formulate the virtual forces, we propose to use fuzzy logic system to control the sensor movement. To save battery life, we refer our appoach to open-loop control systems, in which the control action is independent of the physical system output, e.g. feedback control. Each individual mobile sensor uses a FLS to determine its moving distance and direction, and uncertain exhaustive move and oscillation is efficiently avoided and fast deployment is achieved. The entire network as a result, survives for longer lifetime and the quality of communication in terms of outage probability is greatly ameliorated as the topology approaches uniformity. A concept of coherence time is introduced for the purpose of synchronization among sensors.

The rest of this paper is organized as follows. In section 2, we briefly review the basic concept of fuzzy logic systems. Section 3 details the fuzzy logic systems (FLSs) designed for sensor deployment. Simulation and discussion are presented in Section 4. Section 5 concludes with a summary. Fenton-

Wilkinson method to tackle the outage problem is expatiated in appendix.

2 Overview of Fuzzy Logic Systems

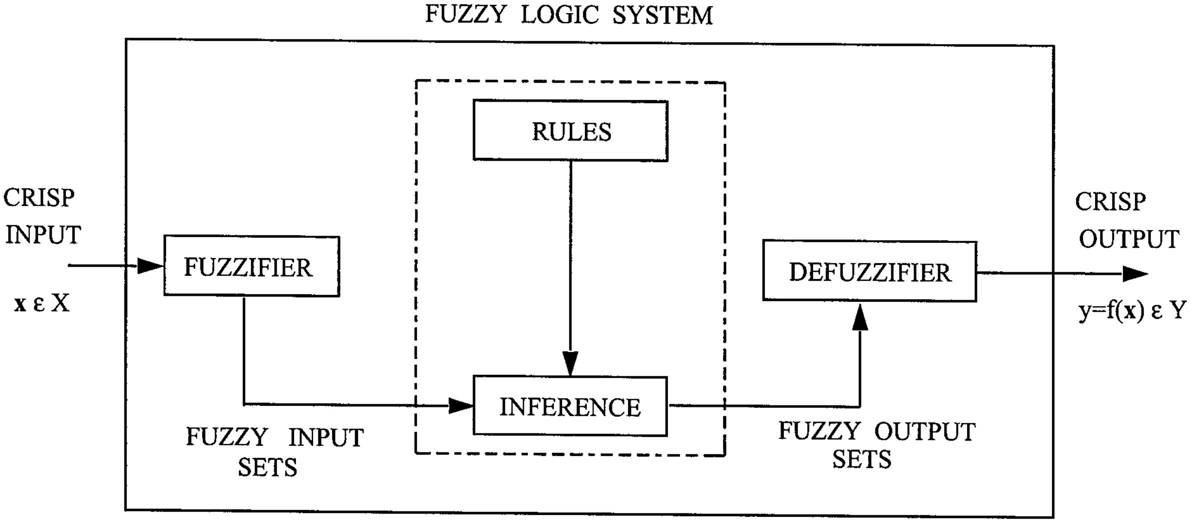


Figure 1: The structure of a fuzzy logic system.

Figure 1 shows the structure of a fuzzy logic system (FLS) [6]. When an input is applied to a FLS, the inference engine computes the output set corresponding to each rule. The defuzzifer then computes a crisp output from these rule output sets. Consider a p-input 1-output FLS, using singleton fuzzification, *center-of-sets* defuzzification [7] and "IF-THEN" rules of the form [8]

$$R^l : \text{IF } x_1 \text{ is } \mathrm{F}_1^l \text{ and } x_2 \text{ is } \mathrm{F}_2^l \text{ and } \cdots \text{ and } x_p \text{ is } \mathrm{F}_p^l, \text{ THEN } y \text{ is } \mathrm{G}^l.$$

Assuming singleton fuzzification, when an input $\mathbf{x}' = \{x_1', \ldots, x_p'\}$ is applied, the degree of firing corresponding to the lth rule is computed as

$$\mu_{\mathrm{F}_1^l}(x_1') \star \mu_{\mathrm{F}_2^l}(x_2') \star \cdots \star \mu_{\mathrm{F}_p^l}(x_p') = \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i') \tag{1}$$

where $\star$ and $\mathcal{T}$ both indicate the chosen t-norm. There are many kinds of defuzzifiers. In this paper, we focus, for illustrative purposes, on the center-of-sets defuzzifier [7]. It computes a crisp output for the FLS by first computing the centroid, c_{G^l}, of every consequent set G^l, and, then computing a weighted average of these centroids. The weight corresponding to the lth rule consequent centroid is the degree of firing associated with the lth rule, $\mathcal{T}_{i=1}^{p}\mu_{\mathrm{F}_i^l}(x_i')$, so that

$$y_{cos}(\mathbf{x}') = \frac{\sum_{l=1}^{M} c_{\mathrm{G}^l}\mathcal{T}_{i=1}^{p}\mu_{\mathrm{F}_i^l}(x_i')}{\sum_{l=1}^{M} \mathcal{T}_{i=1}^{p}\mu_{\mathrm{F}_i^l}(x_i')} \tag{2}$$

where M is the number of rules in the FLS.

3 FLSs for Distributed Sensor Deployment

3.1 Assumptions and Notations

- Sensor field is denoted by a two-dimensional grid. Detection and communication is modeled as a circle on this grid.

- Coverage discussed in this paper is grid coverage. A grid point is covered when at least one sensor covers this point.

- A sensor can detect or sense any event within its sensing range, denoted by Rs. Coverage is determined based on Rs.

- Two sensors within their communication range, denoted by Rc can communicate with each other.

- Neighbors of a sensor are defined as nodes within its communication range.

3.2 Fuzzy Logic System Design

Fuzzy logic system is well known to be able to handle uncertainty and ambiguity. Practically not all uncertainty is random. Some forms of uncertainty are non-random and hence not suited to treatment or modeling by probability theory. Fuzzy set theory is a marvelous tool for modeling the kind of uncertainty associated with vagueness, or with a lack of information regarding a pariticular element of the problem at hand. Upon concerning the distributed sensor deployment, the moving distance and direction of each sensor are distributed and full of uncertainty which can barely be described by some random distribution. Fuzzy logic system is well known as model free. Their membership functions are not based on statistical distributions. Therefore we propose to apply fuzzy logic system to the distributed sensor deployment problem. Each sensor makes fully distributed decision on its movement based on FLS.

Our algorithm starts with random deployment. In the initial condition, a given number of sensors are randomly deployed in a square sensor field. We have made the following assumptions:

- All sensor nodes are peer to peer.
- Sensor nodes have certain mobility and capabilities of computing, detection and communication.
- Sensor node knows its location information.
- Sensors are synchronized by coherence time. One-time move is made within each coherence period.

Every sensor in the network needs to know which direction he is heading to and how far he can reach in the next step. Thus two critical procedures are considered in our algorithm:

- Determine the next-step move distance for each sensor.
- Determine the next-step move direction for each sensor.

The next-step move distance is hard to determine. Too small or big move distance each step consumes the network more time and energy to get stable deployment. Excessive move and oscillation is unavoidable in previous work with no fuzzy control. In this paper, we design a fuzzy logic system to determine the next-step move distance for each sensor.

We collect the knowledge for deployment problem based on the following two antecedents:

Antecedent 1. Number of neighbors of each sensor.

Antecedent 2. Average Euclidean distance between sensor node and its neighbors

The linguistic variables to represent the number of neighbors for each sensor are divided into three levels: *high*, *moderate* and *low*; and those to represent the average Euclidean distance between sensor node and its neighbors are divided into three levels: *far*, *moderate* and *near*. The consequent - the shift distance normalized by sensing range Rs is divided into three levels: *far*, *moderate* and *near*. Table 1 summaries the rules and consequents.

Table 1: Fuzzy Rules and Consequent

$Antecedent1$	$Antecedent2$	$Consequent$
Low	$Near$	$Moderate$
Low	$Moderate$	$Near$
Low	Far	$Near$
$Moderate$	$Near$	Far
$Moderate$	$Moderate$	$Moderate$
$Moderate$	Far	$Near$
$High$	$Near$	Far
$High$	$Moderate$	$Moderate$
$High$	Far	$Moderate$

An ideal sensor deployment should have uniform distribution for better coverage. But in random deployment, coverage uniformity is hardly to achieve initially. In sensor network composed of mobile sensors, each sensor detects the number and location of its neighbors and decides if its neighborhood is sparse or over crowded. If it is over crowded, the sensor makes decision using FLSs to shift a certain distance away from its current location. If it has very few and sparse neighbors, the sensor might stand still or shift a little but staying more closer to its current location.

One example of rules is as follows:

IF the number of neighbors of sensor i is *high* and average Euclidean distance between sensor i and its neighbors is *moderate*, THEN the normalized shift distance of sensor i should be *high*.

We set up 9 rules for this FLS because every antecedent has 3 fuzzy sub-sets and there are 2 antecedents. Trapezoidal membership functions (MFs) are used to represent *high*, *low*, *far* and *near* and triangle MFs to represent *moderate*. Two antecedents are normalized to the range $[0, 10]$. We show these membership functions in Figure 2.

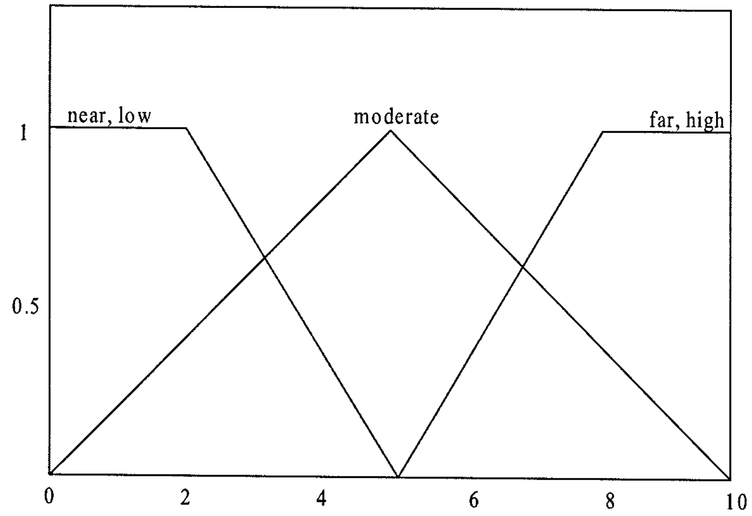


Figure 2: Antecedent Membership Function

Applying center-of-sets defuzzification [7], for every input (x_1, x_2), the output is computed using

$$y_{(x_1,x_2)} = \frac{\sum_{l=1}^{9} c_{G^l}\mu_{F_1^l}(x_1)\mu_{F_2^l}(x_2)}{\sum_{l=1}^{9} \mu_{F_1^l}(x_1)\mu_{F_2^l}(x_2)} \tag{3}$$

Repeating these calculations for $\forall x_i \in [0, 10]$,we obtain a control surface $y(x_1,\ x_2)$ as shown in Figure 3.

The concept of control surface, or decison surface, is central in fuzzy logic systems. It describes the dynamics of the controller and is generally a time-varying nonlinear surface. From figure 3, we can see that although the number of neighbors for a certain sensor is high, the move distance can be nearer than some sensor with fewer "crowded" neighbors, i.e. very close average Euclidean distance between the sensor and its neighbors. With the assist of control surface, the next-step move distance can be determined.

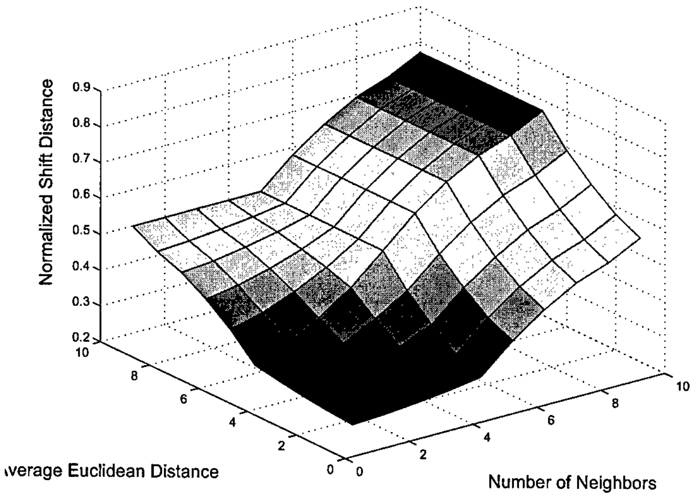


Figure 3: Control Surface

Comparing to move distance, the next-step move direction is much easier to decide. Coulomb's law in physics becomes a useful tool to tackle the problem. For instance, assume sensor i has 2 neighbors in its communication range as shown in Figure 4.

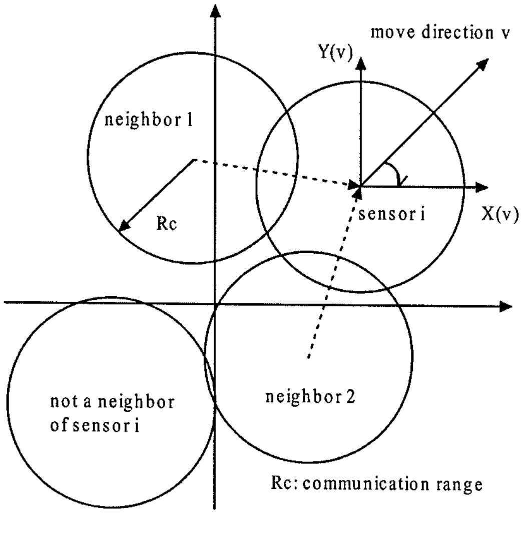


Figure 4: Next Step Move Direction

The coordinate of sensor i is denoted as $C_i = (X_i,Y_i)$.

The next-step move direction of sensor i could be represented as follows:

$$\vec{v} = \sum_{j=1}^{2} \frac{\vec{C_j} - \vec{C_i}}{|\vec{C_j} - \vec{C_i}|^2} \tag{4}$$

$$tan(\alpha) = \frac{Y_{(\vec{v})}}{X_{(\vec{v})}} \tag{5}$$

After getting distance and direction (angle α) , sensor i clearly knows his next-step move information. In order to prolong the battery life of each individual sensor, we introduce a coherence time as the duty cycle during which the changes of two antecedents can be ignored. Sensors are put into idle or sleep mode if within the coherence time, the information of neighbors remains unchanged.

4 Simulation and Discussion

We investigate various number of sensors deployed in a field of 10×10 square kilometers area. We assume each sensor is equipped with an omni antenna to carry out the task of detection and communication. Evaluation of our scheme follows three criteria: field coverage, outage probability and convergence. Results are averaged over 200 Monte Carlo simulations.

Figure 5 shows at 1 kilometer sensing range (R_s=1km) and 2 kilometer communication range (R_c=2km), the coverage of the initial random deployment and the one after using FLSs. The FLSs scheme could improve the network coverage by 20% - 30% in average. As 60 sensors are deployed, the coverage approaches to 100% after FLSs scheme is implemented.

Figure 6 gives the results when R_s=2km and R_c=4km, the coverage comparison between random deployment and fuzzy optimization algorithm. In the case when 20 sensors are deployed, the coverage after random deployment is initialy around 85%. After FLSs are used, the coverage reaches approximate 98%. The coverage is dramatically improved in the low density network. On the other hand, the above two figures indicate that instead of deploying large amount of sensors, the desired field coverage could also be achieved with fewer sensors.

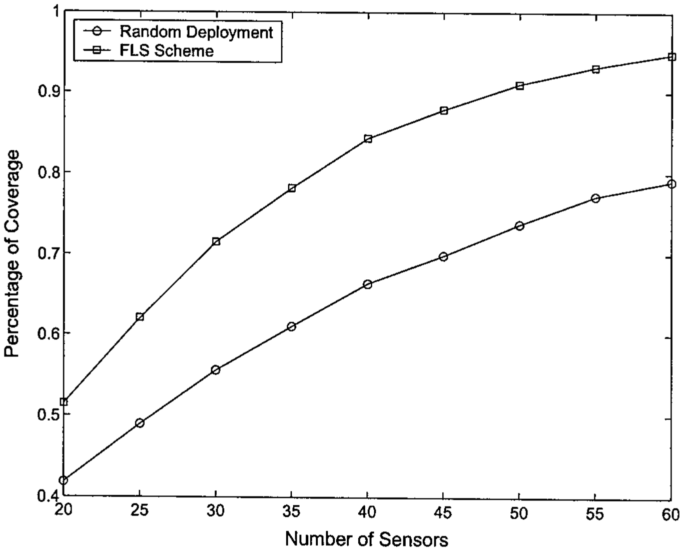


Figure 5: Coverage vs. Number of Nodes (Rc=2,Rs=1)

In celluar radio systems the radio link performance is usually limited by interference rather than noise, therefore, the probability of outage due to co-channel interference is of primary concern. Measurements [9] have shown that at any value of $d_{i,j}$ (the Euclidean distance between sensor i and sensor j), the path loss $PL(d_{i,j})$ is random and distributed log-normally (normal in dB) about the mean distance dependent value. That is:

$$PL(d_{i,j})[dB] = \overline{PL}(d_{i,j}) + X_\sigma = \overline{PL}(d_0) + 10nlog(\frac{d_{i,j}}{d_0}) + X_\sigma \tag{6}$$

and

$$P_r(d_{i,j})[dBm] = P_t[dBm] - PL(d_{i,j})[dB] \tag{7}$$

where X_σ is a zero-mean Gaussian distribution random variable (in dB) with standard deviation σ (also in dB).

The log-normal distribution describes the random *shadowing* effects on the propagation path which implies that measured signal levels at certain distance have a Gaussian (normal) distribution about the distance-dependent mean and standard deviation σ. Since $PL(d_{i,j})$ follows normal

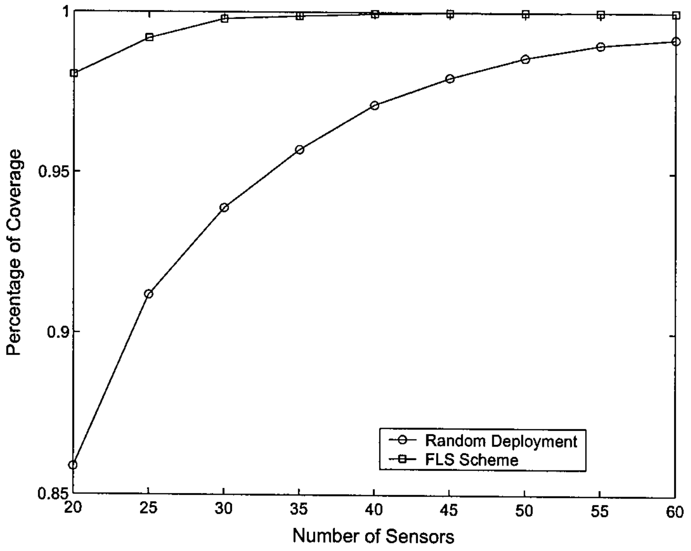


Figure 6: Coverage vs. Number of Nodes (Rc=4,Rs=2)

distribution, so is $P_r(d_{i,j})$, and the Q function may be used to determine the probability that the received signal level will exceed (or fall below) a particular level.

The probability that the received signal level will exceed a certain value γ can be calculated from the cumulative density function as

$$P_r[P_r(d_{i,j}) > \gamma] = Q\left(\frac{\gamma - \overline{P_r}(d_{i,j})}{\sigma}\right) \tag{8}$$

For sensor i with N neighbors, if sensor i acts as the destination node during one communication, the signal to interference ratio (SIR) is represented as:

$$SIR(i) = \frac{P_r(d_{i,j})}{\sum_{k=1}^{N} P_r(d_{i,k})}, k \neq j \tag{9}$$

The denominator denoting the effect of co-channel interference is a sum of $N-1$ log-normal signals. Evaluating the outage probability requires the probability distribution of the interference power. There is no known exact expression for the probability distribution for the sum of log-normal random variables, but various authors have derived several approaches which approximate the sum of log-normal random variables by another log-normal random variable.

In this paper, we use Fenton-Wilkinson method [10]. The co-channel interference can now be approximated by one log-normal random variable. SIR(in dB) as a result follows log-normal distribution as well. We expatiate the Fenton-Wilkinson method in the appendix. Results of outage probability are presented in figure 7.

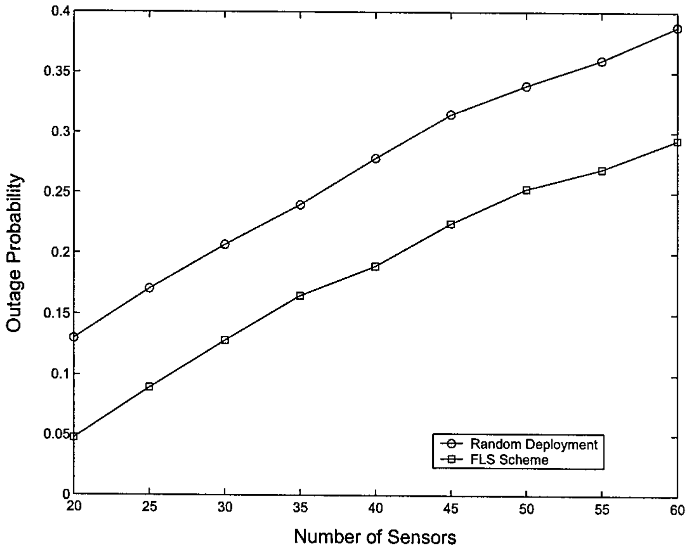


Figure 7: Outage Probability vs. Number of Sensors (Rc=4,Rs=2)

From Figure 7, we can see that the FLSs scheme successfully reduced the outage probability by nearly 30% which implies a higher probability that the received signal level will exceed the SIR threshold. The quality of communications then can be greatly ameliorated.

The performance of the FLSs scheme can also be evaluated in terms of convergence speed. We demonstrate the coverage at each iteration e.g. coverage at the ith iteration is $Cov(i)$ for different number of sensors and the mean square errors (MSE) between adjacent iterations shown as below.

$$MSE(i) = \frac{[Cov(i) - Cov(i-1)]^2}{Cov(i-1)^2} i = 1, ..., iteration \tag{10}$$

Figure 8 and 9 show different number of sensors have approximate same convergence speed. The coverage increases vastly within the first 2 to 3 iterations. After that coverage approaches stable for all different number of sensors. Figure 9 shows at R_s=1km and R_c=2km the coverage

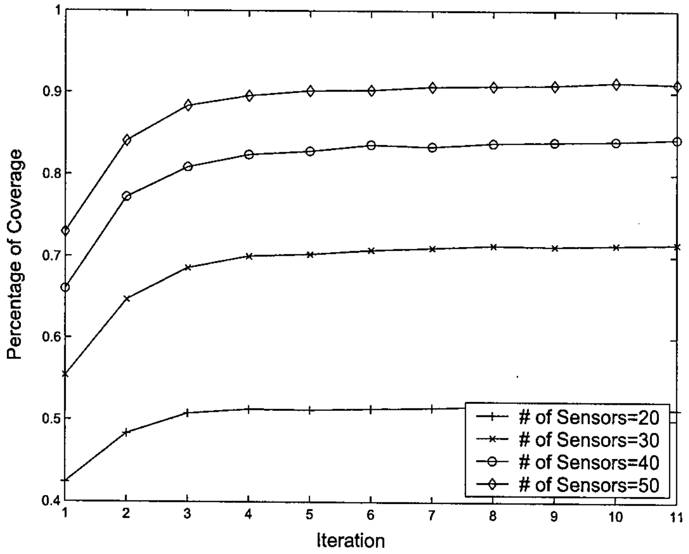


Figure 8: Coverage vs.Interation (Rc=2,Rs=1)

mean square errors between the adjacent iterations. The results also validate that the convergence of the FLSs scheme is independent of the number of sensors to be deployed.

Comparing to other algorithm such as distributed self spreading algorithm [4] which takes over 20 termination times and 10 oscillations and status limit to achieve the similar coverage improvement, the FLSs scheme is much simpler to implement and outperforms in a fast and guaranteed convergence.

5 Conclusions

In this paper, we proposed a new sensor deployment strategy based on fuzzy logic system. Our approach has a great advantage to deal with the uncertainty in distributed sensor deployment which is particularly useful when emergency rescue or redeployment over hostile situation is needed. We believe that in an energy constraint wireless sensor network, fast and efficient deployment strategy is a necessity to save battery power and extend network lifetime. Our FLSs scheme is capable to model all distributed sensor deployment with a fuzzy logic system. The network coverage and quality of communication in term of outage probablity are greatly improved as a result. Moreover,

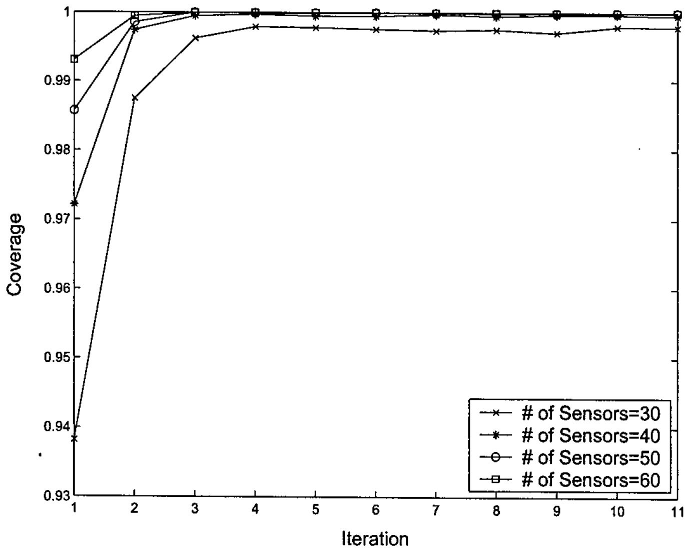


Figure 9: Coverage vs.Interation (Rc=4,Rs=2)

the FLSs scheme brings the whole network to a stable and optimal deployment very soon which will significantly reduce the energy consumption. Our future work will focus on modeling the random deployment with some existing pattern so that the energy consumption can be further studied in the deployment problem.

Acknowledgment

This work was supported by the Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466, "Energy Efficient Wireless Sensor Networs for Future Combat System Using Fuzzy Logic".

Appendix

A Multiple Log-Normal Interferers

Consider the sum of N_I log-normal random variables

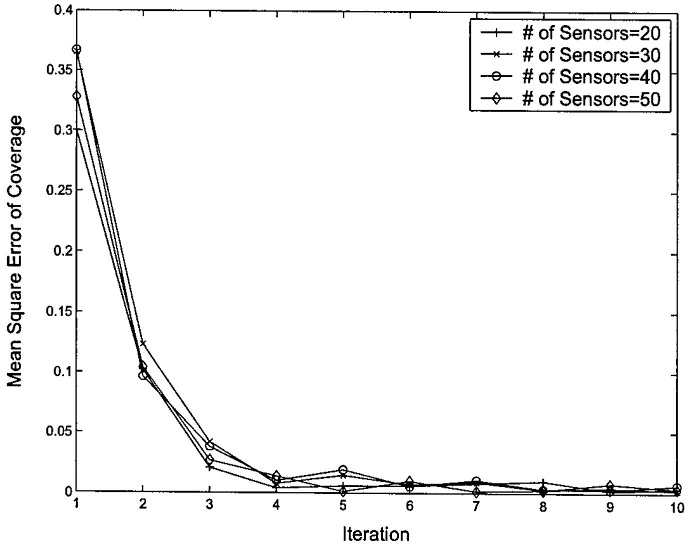


Figure 10: MSE vs. Coverage (Rc=2,Rs=1)

$$I=\sum_{k=1}^{N_I}\Omega_k=\sum_{k=1}^{N_I}10^{\Omega_{k(dBm)}/10} \tag{11}$$

where the $\Omega_{k(dBm)}$ are Gaussian random variables with mean $\mu_{\Omega_{k(dBm)}}$ and variance $\sigma^2_{\Omega_k}$, and the $\Omega_k = 10^{\Omega_{k(dBm)}/10}$ are the log-normal random variables. Unfortunately, there is no known closed form expression for the probability density function (pdf) of the sum of multiple ($N_I \geq 2$) log-normal random variables. However, there is a general consensus that the sum of independent log-normal random variables can be approximated by another log-normal random variable with appropriately chosen parameters. That is,

$$I=\sum_{k=1}^{N_I}10^{\Omega_{k(dBm)}/10}\approx 10^{Z_{(dBm)}/10}=\hat{I} \tag{12}$$

where $Z_{(dBm)}$ is a Gaussian random variable with mean $\mu_{Z(dBm)}$ and variance σ^2_Z. The problem is to determine $\mu_{Z(dBm)}$ and variance σ^2_Z in terms of the $\mu_{\Omega_{k(dBm)}}$ and variance $\sigma^2_{\Omega_k}$, $k = 1, ..., N_I$. Several methods have been suggested in the literature to solove this problem including those by Fenton, Schwartz and Yen, and Farley. Each of these methods provides varying degrees of accuracy over specified ranges of the shadow standard deviation σ_Ω, the sum I, and the number of interferes

N_I.

B Fenton-Wilkinson Method

With the Fenton-Wilkinson method, the mean $\mu_{Z(dBm)}$ and variance σ_Z^2 of $Z_{(dBm)}$ are obtained by matching the first two moments of the sum I with the first two moments of the approximation $\hat{I}$. To derive the appropriate moments, it is convenient to use natural logarithms. We write

$$\Omega_k = 10^{\Omega_{k(dBm)}/10} = e^{\epsilon\Omega_{k(dBm)}} = e^{\hat{\Omega}_k} \tag{13}$$

where $\epsilon = (ln10)/10 = 0.23026$ and $\hat{\Omega}_k = \epsilon\Omega_{k(dBm)}$. Note that $\mu_{\hat{\Omega}_k} = \epsilon\mu_{\Omega_{k(dBm)}}$ and $\sigma^2_{\hat{\Omega}_k} = \epsilon^2\sigma^2_{\Omega_k}$. The nth moment of the log-normal random variable Ω_k can be obtained from the moment generating function of the Gaussian random variables $\hat{\Omega}_k$ as

$$E[\Omega_k^n] = E[e^{n\hat{\Omega}_k}] = e^{n\mu_{\hat{\Omega}_k}+(1/2)n^2\sigma^2_{\hat{\Omega}_k}} \tag{14}$$

To find the appropriate moments for the log-normal approximation we can use (14) and equate the first two moments on both sides of the equation

$$I = \sum_{k=1}^{N_I} e^{\hat{\Omega}_k} \approx e^{\hat{Z}} = \hat{I} \tag{15}$$

where $\hat{Z} = \epsilon Z_{(dBm)}$. For example, suppose that $\hat{\Omega}_k$, $k = 1, ..., N_I$ have mean $\mu_{\hat{\Omega}_k}$, $k = 1, ..., N_I$ and identical variances $\sigma^2_{\hat{\Omega}}$. Identical variances are often assumed because the standard deviation of log-normal shadowing is largely independent of the radio path length. Equating the means on both sides of (15)

$$\mu_I = E[I] = \sum_{k=1}^{N_I} E[e^{\hat{\Omega}_k}] = E[e^{\hat{Z}}] = E[\hat{I}] = \mu_{\hat{I}} \tag{16}$$

gives the result

$$\left(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}}\right) e^{(1/2)\sigma^2_{\hat{\Omega}}} = e^{\mu_{\hat{Z}}+(1/2)\sigma^2_{\hat{Z}}} \tag{17}$$

Likewise, we can equate the variances on both sides of (15) under the assumption that the $\hat{\Omega}_k$, $k = 1, ..., N_I$ are independent

$$\sigma_I^2 = E[I^2] - \mu_I^2 = E[\hat{I}^2] = \sigma_{\hat{I}}^2 \tag{18}$$

giving the result

$$\left(\sum_{k=1}^{N_I} e^{2\mu_{\hat{\Omega}_k}}\right) e^{\sigma^2_{\hat{\Omega}}}(e^{\sigma^2_{\hat{\Omega}}} - 1) = e^{2\mu_{\hat{Z}}} e^{\sigma^2_{\hat{Z}}}(e^{\sigma^2_{\hat{Z}}} - 1) \tag{19}$$

By squaring each side of (17) and dividing each side of resulting equation by the respective side of (19) We can solve for σ^2_{hatz} in terms of the known values of $\mu_{\hat{\Omega}_k}$, $k = 1, ..., N_I$ and $\sigma^2_{\hat{\Omega}}$. Afterwards, $\mu_{\hat{Z}}$ can be obtained from (17). This procedure yields the following solution:

$$\mu_{\hat{Z}} = \frac{\sigma^2_{\hat{\Omega}} - \sigma^2_{\hat{Z}}}{2} + ln\left(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}}\right) \tag{20}$$

$$\sigma^2_{\hat{Z}} = ln\left((e^{\sigma^2_{\hat{\Omega}}} - 1)\frac{\sum_{k=1}^{N_I} e^{2\mu_{\hat{\Omega}_k}}}{(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}})^2} + 1\right) \tag{21}$$

Finally, $\mu_{Z(dBm)} = \epsilon^{-1}\mu_{\hat{Z}}$ and $\sigma^2_Z = \epsilon^{-2}\sigma^2_{\hat{Z}}$.

The accuracy of this log-normal approximation can be measured in terms of how accurately the first two moments of $I_{(dB)} = 10log_{10}I$ are estimated, and how well the cumulative distribution function (cdf) of $I_{(dB)}$ is described by a Gaussian cdf. In problems relating to the co-channel interference outage in celluar radio systems, we are usually interested in the tails of both the complementary distribution function (cdfc) $F_I^C = P(I \geq x)$ and the cdf $F_I(x) = 1 - F_I^C = P(I < x)$. In this case, we are interested in the accuracy of the approximation

$$F_I(x) \approx P(e^{\hat{Z}} \geq x) = Q\left(\frac{lnx - \mu_{\hat{Z}}}{\sigma_{\hat{Z}}}\right) \tag{22}$$

for large and small values of x. It will be shown later that the Fenton-Wilkinson method can approximate the tails of the cdf and cdfc functions with good accuracy.

References

[1] C. Y. Chong and S. P . Kumar "Sensor Networks: Evolution, Opportunities, and Challenges" *Proc.IEEE*, vol. 91, no. 8, pp. 1247-1256, Aug. 2003.

[2] H. Qi, S. S. Iyengar and K. Chakrabarty "Distributed Sensor fusion - a review of recent research" *Journal of the Franklin Institute*, vol. 338, pp. 655-668, 2001.

[3] Y. Zhou and K. Chakrabarty "Sensor deployment and target localization based on virtual forces" *IEEE Twenty-Second Annual Joint Conference of the Computer and Communications Societies*, vol. 2, pp. 1293-1303, 2003.

[4] N. Heo and P. K. Varshney "A distributed self spreading algorithm for mobile wireless sensor networks" *IEEE International Conference on Wireless Communications and Networking*, vol. 3, pp. 1597 - 1602, 2003.

[5] S. Poduri and G. S. Sukhatme "Constrained coverage for mobile sensor networks" *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 165 - 171,2004

[6] J. M. Mendel "Fuzzy Logic Systems for Engeneering: A Tutorial" *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345 - 377,1995

[7] J. M. Mendel "Uncertain Rule-Based Fuzzy Logic Systems" *Prentice-Hall, Upper Saddle River, NJ*, 2001

[8] E. H. Mamdani, " Applications of fuzzy logic to approximate reasoning using linguistic systems", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 26, no. 12, pp. 1182-1191, 1977.

[9] T. S. Rappaport, "Wireless communications:principles and practice" *Prentice-Hall, Upper Saddle River, NJ*, 2001

[10] G. L. Stuber "Principles of mobile communication" *Boston : Kluwer Academic Publishers,* 2001

Wireless Sensor Network Lifetime Analysis Using Interval Type-2 Fuzzy Logic Systems

Haining Shu and Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016 USA
E-mail: shu@wcn.uta.edu, liang@uta.edu

Abstract—**Prolonging the lifetime of energy constrained wireless sensor networks is a crucial challenge in sensor network research. In this paper, we present a novel approach based on fuzzy logic theory to analyze the lifetime performance of a sensor network. We demonstrate that a type-2 fuzzy membership function(MF), i.e., a Gaussian MF with uncertain variance is most appropriate to model a single node lifetime in wireless sensor networks. In our research, we concern two basic sensor placement schemes: square-grid and hex-grid. Two fuzzy logic systems(FLSs): a singleton type-1 FLS and an interval type-2 FLS are designed to perform lifetime estimation of the entire sensor network. Simulation results show that the interval type-2 FLS in which the antecedent membership functions are modeled as type-2 fuzzy sets outperforms the singleton type-1 FLS when the single node lifetime behaves the nature of Gaussian MFs with uncertain standard deviation.**

I. Introduction

Wireless sensor network represents significant advantages over conventional personnel-rich methods of long term data collection and monitoring. Starting in the 1980s, networked microsensors technology has been widely used in military applications. Examples of these applications include the Cooperative Engagement Capability (CEC), Remote Battlefield Sensor System (REMBASS), Advanced Deployable System (ADS) and Tactical Remote Sensor System (TRSS) [1]. Although a majority of military applications deploy sensor nodes randomly, a number of other applications require the placement of sensors at desired locations. Such placement-friendly sensor networks are developed rapidly for infrastructure security, environment and habitat monitoring, traffic control etc [1]. A real-world 32 node habitat monitoring sensor network system was deployed on a small island off the coast of Maine to study nesting patterns of Petrels [2]. Other applications involve the use of sensors in buildings for environmental monitoring which may include chemical sensing and detection of moisture problems. Structural monitoring [3] and inventory control are some other applications of such networks.

Wireless sensor network consists of certain amount of small and energy constrained nodes. Basic components of sensor node include a single or multiple sensor modules, a wireless transmitter-receiver module, a computational module and a power supply module. Such networks are normally deployed for data collection where human intervention after deployment, to recharge or replace node batteries may not be feasible, resulting in limited network lifetime. Most applications have prespecified lifetime requirements, for instance the application in [2] has a lifetime requirement of at least 9 months. Thus estimation of lifetime of such networks prior to deployment becomes a necessity. Prior works on evaluating lifetime have considered networks where sensor nodes are randomly deployed. [4] gives the upper bound on lifetime that any network with the specified number of randomly deployed nodes, source behavior and energy can reach while [6] discusses the upper bounds on lifetime of networks with cooperative cell based strategies. Network liftime of fixed deployment schemes are recently studied in [7]. Jain and Liang [7] discovers that in wireless sensor networks, a single node lifetime behaves the nature of normal distribution which brings the first light of exploring the network lifetime behavior given the knowledge of nodes lifetimes.

In this paper, we deal with the issue of lifetime analysis and estimation for wireless sensor networks in which the sensor nodes are deployed at desired locations. We propose to apply interval type-2 fuzzy logic systems (FLSs) for network lifetime analysis and estimation. Our approach is entirely different from all prior research. We demonstrate that a type-2 fuzzy membership function(MF), i.e., a Gaussian MF with uncertain variance is most appropriate to model a single node lifetime in wireless sensor networks. Two fuzzy logic systems(FLSs): a singleton type-1 FLS and an interval type-2 FLS are constructed to perform lifetime estimation of the entire sensor network. In our work, we concern two basic sensor placement schemes: square-grid and hex-grid. We believe that these two schemes can serve as basis for evaluating more complex schemes for their lifetime performance prior to deployment and help justify the deployment costs.

The rest of this paper is organized as follows. In section II, we detail two sensor deployment schemes used for this study and the basic concepts on network coverage, connectivity and lifetime. Section III gives an overview of interval type-2 fuzzy logic systmes. In section IV, we demonstrate that a single node lifetime can be modeled with Gaussian MFs. We found that a type-2 Gaussian MF with uncertain variance is most appropriate to model a single node lifetime in wireless sensor networks. We apply this knowledge and design an interval type-2 FLS in section V to analyze network lifetime. A singleton type-1 FLS is constructed as well for performance comparison. Simulation results and discussions are also presented in section V. Section VI concludes this paper.

II. Preliminaries

A. Basic Model and Assumptions

Consider identical wireless sensor nodes placed in a square sensor field of area A. All nodes are deployed with equal energy. Each sensor is capable of sensing events up to a radius r_s, the sensing range. Communication range r_c is defined as the distance beyond which the transmitted signal is received with signal to noise ratio (SNR) below the acceptable threshold level. In this paper, We assume the communication range r_c to be equal to the sensing range r_s.

Direct communication between two sensor nodes is possible only if their distance of separation r is such that $r \leq r_c$. We call such nodes *neighbors*. Communication between a sensor node and its non-neighboring node is achieved via peer-to-peer communication. Thus the maximum allowable distance between two nodes who wish to communicate directly is $r_{max} = r_c = r_s$. A network is said to be deployed with minimum density when the distance between its neighboring nodes is $r = r_{max}$.

B. Placement Schemes

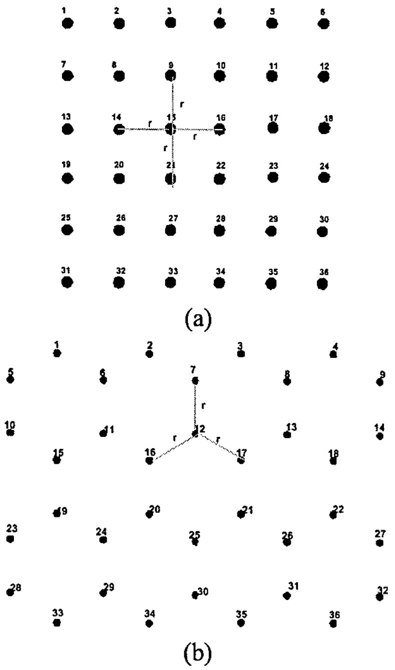
The simplest placement schemes involve regular placement of nodes such that each node in the network has the same number of neighbors. We arrive at two basic placement schemes by considering cases where each sensor nodes has four and three neighbors. This leads us to the *square-grid* and *hex-grid* placement schemes shown in Fig. 1(a) and (b) respectively.

A sensor node placement scheme that uses two neighbors per sensor node has been described in [13]. We believe that these three elementary placement schemes can serve as basis for other placement schemes, because a placement scheme of any complexity can be decomposed into two-neighbor, three-neighbor and four-neighbor groups. Both grids shown have the same number of nodes[1] and nodes in both grids are equidistant from their respective neighbors (with distance of separation r).

C. Coverage and Connectivity

Coverage and connectivity are two important performance metrics of networks and hence a discussion on them becomes imperative before the lifetime of the network can be defined.

Coverage scales the adequacy with which the network covers the sensor field. A sensor with sensing range r_s is said to cover or sense a circular region of radius r_s around it. If every point in the sensor field is within distance r_s from at least one sensor node, then the network is said to provide complete or 100% coverage.



(a)

(b)

Fig. 1. Placement Schemes for a 36 node sensor network: (a) Square-Grid (b) Hex-Grid.

Connectivity scales the adequacy with which nodes are able to communicate with their peers. One of the strengths of sensor networks arises from their ability to aggregate data collected from different sensor nodes. This requires adequate communication between sensor nodes. Any node should be able to communicate with any other node for proper functioning of the network. If a single node gets isolated due to failure of all its neighbors, it will be unable to communicate with the rest of the network. If a large number of nodes fail due to lack of energy, a part of the network may get completely disconnected from the rest.

In our analysis we require the network to provide complete coverage and connectivity. We give equal importance to both parameters and declare the network nonfunctional if either of them falls below their desired levels.

D. Lifetime

The basic definition of lifetime, or more precisely the post-deployment active lifetime of a network is the time measured from deployment until network failure. Based on the levels of coverage and connectivity required to deem a network functional, network failure can be interpreted in different ways. Since only complete coverage and connectivity are acceptable to us, network failure corresponds to the first loss of coverage or connectivity.

In this paper, we concentrate on finding the minimum lifetime of a network, the worst case scenario. Consider the square-grid and the hex-grid deployed with minimum density. Both schemes survive the failure of a single node without loss of either connectivity or coverage however failure of any two neighboring nodes causes loss of coverage and hence network failure as indicated in Figs 2(a) and (b).

Thus the minimum number of node failures that cause

[1] The *Hex-grid* has lower density than the *Square-grid.* With 36 nodes deployed, the network with *Square-grid* covers an area of $25r^2$, and the *Hex-grid* covers an area of $48r^2$, almost double that of the *square grid.*

network failure is two and these two nodes must be adjacent to each other (neighbors). A network may undergo multiple node failures and still be connected and covered if any of the failed nodes are not neighbors. But the absolute minimum number of node failures that can cause network failure is two.

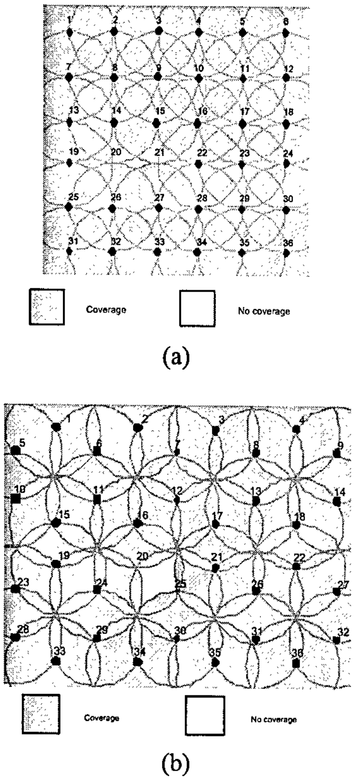


Fig. 2. Loss of coverage due to failure of two neighboring nodes: (a)Square-grid: Failure of nodes 20 and 21 causes loss of coverage. (b)Hex-grid: Failure of nodes 20 and 25 causes loss of coverage.

III. Introduction to Interval Type-2 Fuzzy Logic Systems

Figure 3 shows the structure of a type-2 FLS. It is very similar to the structure of a type-1 FLS [16]. For a type-1 FLS, the *output processing* block only contains the defuzzifier. We assume that the reader is familiar with type-1 FLSs, so that here we focus only on the similarities and differences between the two FLSs.

The fuzzifier maps the crisp input into a fuzzy set. This fuzzy set can, in general, be a type-2 set.

In the type-1 case, we generally have "IF-THEN" rules, where the lth rule has the form "R^l : IF x_1 is F_1^l and x_2 is F_2^l and $\cdots$ and x_p is F_p^l, THEN y is G^l", where: x_is are inputs; F_i^ls are antecedent sets ($i = 1, \ldots, p$); y is the output; and G^ls are consequent sets. The distinction between type-1 and type-2 is associated with the nature of the membership functions, which is not important while forming rules; hence, the structure of the rules remains exactly the same in the type-2 case, the only difference being that now some or all of the sets involved are of type-2; so, the lth rule in a type-2 FLS has the form "R^l : IF x_1 is $\tilde{\mathrm{F}}_1^l$ and x_2 is $\tilde{\mathrm{F}}_2^l$ and $\cdots$ and x_p is $\tilde{\mathrm{F}}_p^l$, THEN y is $\tilde{\mathrm{G}}^l$".

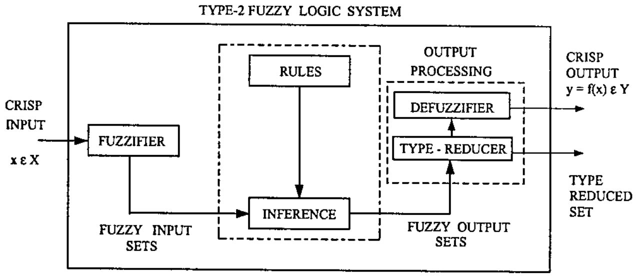


Fig. 3. The structure of a type-2 FLS. In order to emphasize the importance of the type-reduced set, we have shown two outputs for the type-2 FLS, the type-reduced set and the crisp defuzzified value.

In the type-2 case, the inference process is very similar to that in type-1. The inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. To do this, one needs to find unions and intersections of type-2 sets, as well as compositions of type-2 relations.

In a type-1 FLS, the defuzzifier produces a crisp output from the fuzzy set that is the output of the inference engine, i.e., a type-0 (crisp) output is obtained from a type-1 set. In the type-2 case, the output of the inference engine is a type-2 set; so, we use "extended versions" (using Zadeh's Extension Principle [15]) of type-1 defuzzification methods. This extended defuzzification gives a type-1 fuzzy set. Since this operation takes us from the type-2 output sets of the FLS to a type-1 set, we call this operation "type-reduction" and the type-reduced set so obtained a "type-reduced set".

To obtain a crisp output from a type-2 FLS, we can defuzzify the type-reduced set. The most natural way of doing this seems to be by finding the centroid of the type-reduced set; however, there exist other possibilities like choosing the highest membership point in the type-reduced set.

General type-2 FLSs are computationally intensive, because type-reduction is very intensive. Things simplify a lot when secondary membership functions (MFs) are interval sets (in this case, the secondary memberships are either 0 or 1). When the secondary MFs are interval sets, we call the type-2 FLSs "interval type-2 FLSs". In [21], we proposed the theory and design of interval type-2 fuzzy logic systems (FLSs). We proposed an efficient and simplified method to compute the input and antecedent operations for interval type-2 FLSs, one that is based on a general inference formula for them. We introduced the concept of upper and lower membership functions (MFs) and illustrate our efficient inference method for the case of Gaussian primary MFs. We also proposed a method for designing an interval type-2 FLS in which we tune its parameters.

We have developed theory and design methods for the most useful kind of type-2 fuzzy logic system (FLSs), interval type-

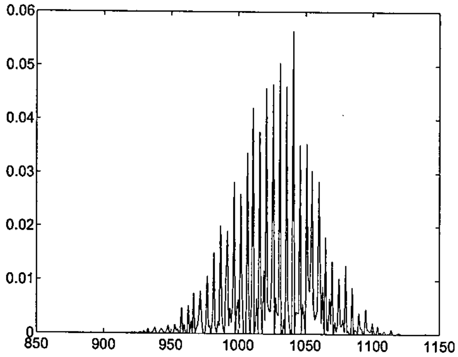


Fig. 4. Single Node Lifetime Distribution

2 FLSs [21], and have applied them to a number of very interesting applications, such as

1) Fading channel equalization [22] and co-channel interference elimination [23]. The channel states in a fading channel or channel with co-channel interferences are uncertain, and we validated that an interval type-2 fuzzy set, Gaussian primary membership function with uncertain mean, can be used to represent such uncertainties.
2) Network video traffic modeling and classification [19]. MPEG variable bit rate (VBR) traffic are very bursty. We validated that the I, P, and B frame sizes are log-normal with fixed mean and uncertain variance, so an interval type-2 fuzzy sets can be used to model the bursty video traffic and an interval type-2 fuzzy logic system with such type-2 fuzzy set are demonstrated performing much better than a Bayesian classifier.
3) Connection admission control for ATM network [20]. Connection admission control is actually a decision making problem. Different factors such as incoming real-time video/audio packet sizes, non-real time packet sizes, the buffer sizes are uncertain. We applied an interval type-2 fuzzy logic to handle these uncertainties, and achieved very good performance.

IV. Modeling Node Lifetime with Gaussian Membership Functions

Wireless sensor nodes are severely energy constrained due to their compact form. To increase the lifetime of sensor networks, hardware design and protocol approaches for different layers must take energy efficiency into account. However, a foundamental question - "what is the nature of sensor network lifetime?" has not been answered yet. Since the lifetimes of individual nodes are not constants but random variables, it follows that the network lifetime is also a random variable. Recent research by Jain and Liang [7] discovered that in a wireless sensor network where the workloads are well-balanced, a single node lifetime behaves the nature of normal distribution as demonstrated in Fig 4.

In this paper, we are first interested in setting up fine membership functions (MFs) for single node lifetime. From the original data of single node lifetime shown in Table I, we decompose the whole data sets into seven segments, and compute the mean μ_i and standard deviation σ_i of node lifetime for each segment, $i = 1, 2, ...7$. The mean μ and standard deviation σ for of the entire data set is also computed. We are also interested to know which value - mean μ_i or standard deviation σ_i varies more. We first normalize the mean μ_i and standard deviation σ_i of each segment using μ_i/μ and σ_i/σ. Then we compute the standard deviation of their normalized values σ_m and σ_{std}. Results are presented in Table I.

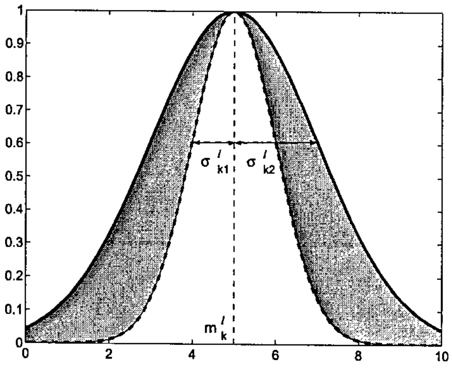


Fig. 5. Type-2 Gaussian MF with uncertain standard deviation

TABLE I

MEAN AND STD VALUES FOR SEVEN SEGMENTS AND THE ENTIRE NODE LIFETIME, AND THEIR NORMALIZED STD

Node Lifetime Data	Mean	Std
Segment 1	1027.4	30.182
Segment 2	1028.9	29.819
Segment 3	1026.3	30.798
Segment 4	1028.7	30.917
Segment 5	1028	29.944
Segment 6	1027	29.975
Segment 7	1027.9	30.306
Entire Data Set	1027.7	30.292
Normalized STD	0.00082783	0.013105

From the last row of Table I, we see that $\sigma_m << \sigma_{std}$ which means standard deviation σ_i varies much more than the mean value μ_i. Therefore we conclude that if the single node lifetime follows normal distribution, it is most appropriate to be modeled as a Gaussian MF with uncertain standard deviation. One example of type-2 Gaussian MF with uncertain standard deviation is shown in Fig 5. This result also justifies the use of the Gaussian MFs to model network lifetime in section V.

V. Network Lifetime Analysis and Estimation Using Interval Type-2 Fuzzy Logic Systems

We are now ready to evaluate the network lifetime using interval type-2 fuzzy logic systems (FLSs). We apply reliability theory to design fuzzy rules. In the following section, we first treat the basics of reliability theory before extracting the

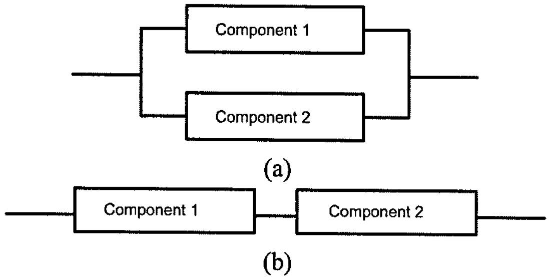


Fig. 6. Reliability block diagrams (*RBD*) for a system of two components. (a)RBD with series connected components. (b)RBD with parallel connected components.

knowledges for fuzzy rules. An interval type-2 FLS is then set up to perform lifetime analysis and estimation.

A. Fuzzy Rules Design Using Reliability Theory

Rules are the heart of a FLS, and may be provided by experts or can be extracted from numerical data. In either case, the rules that we are interested in can be expressed as a collection of IF-THEN statements, e.g.,

IF the noise of the wireless channel is high, THEN the quality of the received signal is poor.

For the lifetime issue studied in this paper, reliability theory provides a feasible method to design fuzzy rules. To understand this, we introduce the reliability block diagram (*RBD*). *RBD* is a graphical representation of the components of the system, and provides a visual representation of the way components are reliability-wise connected. Thus the effect of the success or failure of a component on the system performance can be evaluated.

Consider a system with two components. If this system is such that a single component failure can render the system nonfunctional, then we say that the components are reliability-wise, connected in series. If the system fails only when both its components fail, then we say that the components are reliability-wise connected in parallel. Note that the physical connection between the component may or may not be different from their reliability-wise connection. The *RBD*'s for both cases are given in Figs.6. Any complex system can be realized in the form of a combination of blocks connected in series and parallel.

In our analysis, the wireless sensor network is the system under consideration and the sensor nodes are the components of the system. We demonstrate bellow how the knowledge is extracted for fuzzy rules design referring to the two examples in Fig.6.

Example 1: Set up Fuzzy Rules for Parallel System in Fig. 6(a)

In the parallel system, the system (network) fails only when both components (sensor nodes) fail. The rules can be set up as one example shown bellow:

IF the remaining battery level of component 1 (sensor node 1) is *low* and the remaining battery level of component 2 (sensor node 2) is *moderate*, THEN the lifetime of the system (network) is *moderate*.

Example 2: Set up Fuzzy Rules for Series system in Fig. 6(b)

In the series system, the system (network) fails when either component fails. The rules can be set up as one example shown bellow:

IF the remaining battery level of component 1 (sensor node 1) is *low* and the remaining battery level of component 2 (sensor node 2) is *moderate*, THEN the lifetime of the system (network) is *low*.

Note that the parallel and series systems are the two basic ways to model two sensor nodes. A wireless sensor network consisting of multiple sensor nodes can be first represented in the reliability block diagram. Fuzzy rules are then set up based on the above examples.

B. Simulation and Discussion

In our simulations, interval type-2 FLSs are constructed separately for square-grid netowrk and hex-grid network. We take the remaining energy level of each individual sensor node as input to the interval type-2 FLS and the output is the estimated network lifetime. As we have discussed in section IV, we choose type-2 Gaussian with uncertain standard deviation as the membership functions for both antecedent and consequent. The linguistic variables to represent the antecedent - remaining energy level are divided into three levels: *high*, *moderate* and *low* and the consequent - estimated network life is divided into five levels: *very high*, *high*, *moderate*, *low* and *very low*.

Simulations are performed for both squre-grid and hex-grid sensor networks. In both cases, 36 nodes are deployed and the distance between neighboring nodes is assumed to be the same. All sensor nodes are initialized with maximum energy level of 10 Joules. We assume the workload of the entire network is well-balanced, therefore energy dissipation of the single sensor node can be figured as linear most of the alive time. Our simulations are based on $N = 600$ lifetime data from the above well-balanced sensor network. The first 300 data are for training and the remaining 300 data are for testing. After training, the rules are fixed and we test the interval type-2 by evaluating the root mean square errors (RMSE) between the defuzzified output of FLS and the real lifetime data. We compare our interval type-2 FLSs with singleton type-1 FLSs. For both FLS schemes, we run 200 Monte-Carlo realizations and for each realization, each FLS is tuned using a simple steepest-descent algorithm for six epochs. Simulation results are averaged over all 200 Monte-Carlo realizations.

1) Square Grid: As defined in Section II-D, the minimum network lifetime is the time to failure of any two neighboring nodes. We know that the failure of any single node does not

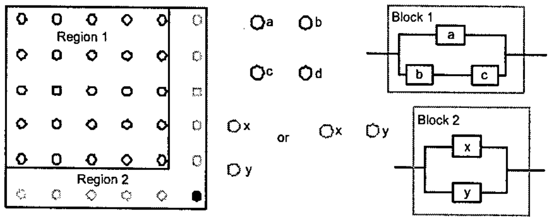


Fig. 7. RBD of a single node in a square grid. Nodes belonging to region-1 are modeled as block-1 and nodes belonging to region-2 are modeled as block-2. The network RBD consists of $(\sqrt{N_{min}} - 1)^2$ block-1's in series with $2(\sqrt{N_{min}} - 1)$ block-2's

cause network failure. The failure of any node coupled with the failure of any of its neighbors causes network failure. Using this definition we build the RBD for the square-grid as shown in Fig 7.

Fig 7 shows the *RBD* block for a single node in the network. A node can be modeled in two ways depending on its position in the sensor field. This distinction based on its position is made due to a simple observation that nodes at the right edge of the sensor field (region-2) do not have any right neighbor (node b) as opposed to nodes in region-1. Also, nodes at the bottom edge of the sensor field (region-2) do not have a bottom neighbor (node c) as opposed to the nodes in region-1. Note that as every node in a square-grid, node a has four neighbors, but its relationship with only two neighbors is modeled in its *RBD* block. This is because the relationship with the other two neighbors will be modeled when their *RBD* blocks are constructed. If this is not followed then the relationship between every node-neighbor pair will be modeled twice.

In this square grid network shown in Fig 7, we classify three antecedents based on the *RBD* of block-1:

- The remaining battery level of node a.
- The minimum remaining battery level of node b and c.
- The remaining battery level of node d.

We set up 27 rules for this FLS because every antecedent has 3 fuzzy sub-sets and there are 3 antecedents.

Two antecedents are chosen based on the *RBD* of block-2 in Fig 7 and total 9 rules are constructed in this case.

- The remaining battery level of node x.
- The remaining battery level of node y.

Let N_{min} be the number of sensor nodes required to be deployed with minimum density. The network *RBD* consists of $(\sqrt{N_{min}} - 1)^2$ block-1's and $2(\sqrt{N_{min}} - 1)$ block-2's in series, the whole square grid network can actually be decomposed into multiple blocks serial connected together and the method of setting up rules can be applied as well. Simulation results of RMSE is shown in Fig. 8.

2) Hex-Grid: The analysis for the hex-grid is carried out on the same lines as that of the square-grid. Fig 2 (b) shows that as in the case of a square grid, two neighboring node failures cause network failure. The RBD block of a single node is shown in Fig. 9.

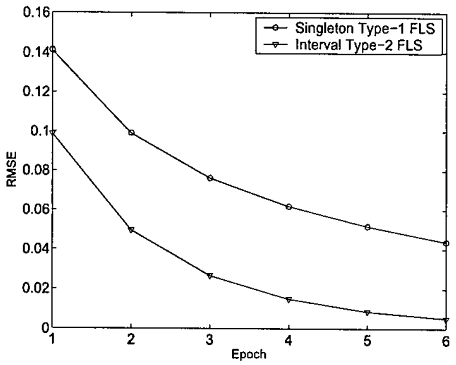


Fig. 8. Square-Grid: The RMSE(for the test data) for two FLS approaches averaged over 200 Monte-Carlo realization

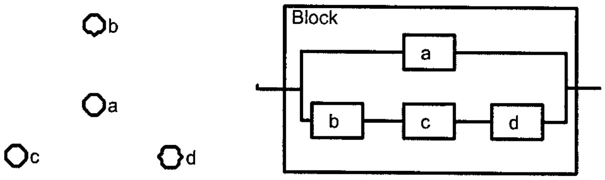


Fig. 9. RBD block for a single node in the Hex-grid: The network RBD consists of $N/2$ such blocks in series.

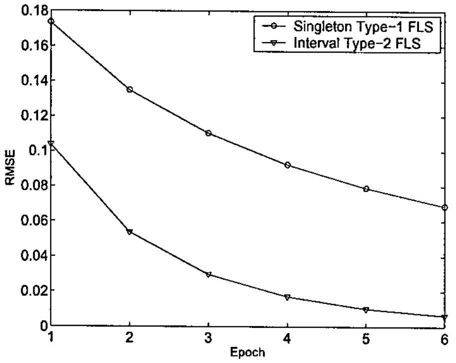


Fig. 10. Hex-Grid: The RMSE(for the test data) for two FLS approaches averaged over 200 Monte-Carlo realization

Since the relation between a node and all of its neighbors is modeled by its corresponding *RBD* block, the *RBD* block's for the neighbors is not constructed as this causes the relationship between the nodes to be considered twice. In this case, we classify two antecedents and construct 9 rules.

- The remaining battery level of node a.
- The minimum remaining battery level of node b, c and d.

Since $N_{min}/2$ such blocks connected in series represent the network, the whole hex grid network can be decomposed the same way as in square grid networks. Simulation results of RMSE is shown in Fig. 10.

Observe Fig.8 and Fig.10, interval type-2 FLSs outperform singleton type-1 FLSs. This results shows that the interval type-2 FLSs are more feasible for real-time energy estimation.

VI. Conclusion and Future Works

In this paper, we describe a new method based on fuzzy logic theory to analyze and estimate the network lifetimes for wireless sensor networks. Our approach is illuminated by the discovery that a single node lifetime behaves the nature of normal distribution. However,we deem that if the single node lifetime follows normal distribution, it is most appropriate to be modeled as a Gaussian MF with uncertain standard deviation. We then set up the interval type-2 FLSs for energy estimation and evaluate their performance using real lifetime data. Simulation results justified the feasibility of applying type-2 FLSs into wireless sensor network lifetime analysis. Interval type-2 FLSs provides a way to handel knowledge uncertainty. We believe that our approaches opens up a new vision for research on sensor network lifetimes.

Our future work will focus on lifetime evaluation under the circumstances that the task scheduling is variable and how the estimated network lifetime could be used to accomodate the scheduling change.

Acknowledgment

This work was supported by the U.S. Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466, "Energy Efficient Wireless Sensor Networs for Future Combat System Using Fuzzy Logic".

References

[1] C.Y. Chong, S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges" *Proc. IEEE* , vol 91, no. 8 , Aug 2003, pp. 1247 -1256

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, "Wireless Sensor Networks for Habitat Monitoring" *Proc. WSNA'02* Atlanta, Georgia, Sep 28, 2002.

[3] V. A. Kottapalli, A. S. Kiremidjian, J. P. Lynch, Ed Carryer, T. W. Kenny, "Two-tired wireless sensor network architecture for structural health monitoring" *Proc. SPIE* San Diego, CA, Mar 2003.

[4] M. Bhardwaj, T. Garnett, A. Chandrakasan, "Upper Bounds on the Lifetime of Sensor Networks" *Proc. IEEE International Conference on Communications,* pp.785-790, 2001.

[5] M. Bhardwaj, A. P. Chandrakasan, "Bounding the Lifetime of Sensor Networks via Optimal Role Assignments" *Proc. INFOCOM 2002* pp:1587 - 1596, vol.3

[6] D. M. Blough, P. Santi, "Investigating Upper Bounds on Network Lifetime Extension for Cell-Based Energy Conservation Techniques in Stationary Ad Hod Networks" *Proc. MOBICOM'2002* Atlanta, Georgia, Sep 2002

[7] E. Jain and Q. Liang, "Sensor placement and lifetime of wireless sensor networks: theory and performance analysis," *Sensor Network Operations,* edited by S. Phoha, T. F. La Porta, and C. Griffin, IEEE Press, Piscataway, NJ, 2004.

[8] N. N. Karnik and J. M. Mendel, *An Introduction to Type-2 Fuzzy Logic Systems,* October 1998, USC Report, http://sipi.usc.edu/∼mendel/report.

[9] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems", *IEEE Trans. Fuzzy Systems*, vol. 7, no. 6, pp. 643-658, Dec. 1999.

[10] N. N. Karnik and J. M. Mendel, "Type-2 Fuzzy Logic Systems : Type-Reduction", presented at the *1998 IEEE SMC Conference*, San Diego, CA, October.

[11] N. N. Karnik, J. M. Mendel, and Q. Liang, "Centroid of a type-2 fuzzy set," *Information Sciences.* vol.132, pp.195-220, Feb, 2001,

[12] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, May/June 1993.

[13] K. Kar, S. Banerjee, "Node Placement for Connected Coverage in Sensor Networks" *Extended Abstract. Proc. WiOpt 2003* Sophia-Antipolis, France, March 2003.

[14] S. S. Dhillon, K. Chakrabarty, S. S. Iyengar, "Sensor Placement for Grid Coverage under Imprecise Detections," FUSION, 2002.

[15] L. A. Zadeh, " The concept of a linguistic variable and its application to approximate reasoning - I," *Information Sciences*, vol. 8, pp. 199-249, 1975.

[16] J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proc. of the IEEE*, vol. 83, no. 3, pp. 345-377, March 1995.

[17] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, USA, 1980.

[18] E. Hisdal, "The IF THEN ELSE statement and interval-valued fuzzy sets of higher type," *Int'l. J. Man-Machine Studies*, vol. 15, pp. 385-455, 1981.

[19] Q. Liang and J. M. Mendel, "MPEG VBR video traffic modeling and classification using fuzzy techniques," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 1, pp.183-193, Feb 2001.

[20] Q. Liang, N. Karnik, and J. M. Mendel, " Connection admission control in ATM network using survey-based type-2 fuzzy logic systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C,* vol. 30, no. 3, pp. 529-539, August 2000.

[21] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535-550, Oct 2000.

[22] Q. Liang and J. M. Mendel, "Equalization of time-varying nonlinear channels using type-2 fuzzy adaptive filters," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 5, pp. 551-563, Oct 2000.

[23] Q. Liang and J. M. Mendel, " Overcoming time-varying co-channel interference using type-2 fuzzy adaptive filters", *IEEE Transactions on Circuits and Systems, II*, vol. 47, no. 12, pp. 1419-1428, Dec 2000.

[24] L.-X. Wang and J. M. Mendel, " Fuzzy basis functions, universal approximation, and orthogonal least squares learning, " *IEEE Trans. on Neural Networks*, vol. 3, pp. 807-814, Sept. 1992.

[25] D. Kececioglu, "Reliability Engineering Handbook" *Volume 1 and 2,* Prentice Hall, Ney Jersey 1991.

[26] L. M. Leemis, "Reliability: Probabilistic Models and Statistical Methods," Prentice- Hall, 1995

[27] Life Data Analysis Reference. [Online] Available: http://www.weibull.com/lifedatawebcontents.htm

[28] A. Papoulis, S. U. Pillai "Probability, Random Varibales and Stochastic Processes, " *4th ed,* McGraw-Hill, New York 2002.

[29] B. Healy, "The Use of Wireless Sensor Networks for Mapping Environmental Conditions in Buildings" *ASHRAE Seminar, July 2 2003* Available Online: http://www.nist.gov/tc75/ASHRAESummer2003SeminarHealy.pdf

Sensed Signal Strength Forecasting for Wireless Sensors Using Interval Type-2 Fuzzy Logic System

Qilian Liang and Lingming Wang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016. USA
E-mail:liang@uta.edu. wang@wcn.uta.edu

***Abstract*— In this paper, we present a new approach for sensed signal strength forecasting in wireless sensors using interval type-2 fuzzy logic system (FLS). We show that a type-2 fuzzy membership function, i.e., a Gaussian MF with uncertain mean is most appropriate to model the sensed signal strength of wireless sensors. We demonstrate that the sensed signals of wireless sensors are self-similar, which means it can be forecasted. An interval type-2 FLS is designed for sensed signal forecasting and is compared against a type-1 FLS. Simulation results show that the interval type-2 FLS performs much better than the type-1 FLS in sensed signal forecasting. This application can be further used for power on/off control in wireless sensors to save battery energy.**

***Index Terms*— Wireless sensors, self-similarity, forecasting, fuzzy logic system.**

I. Introduction

The infusion and maturation of the Micro Mechanical System(MEMS), computations, and wireless communication technologies has advanced the development of wireless sensor networks. A large amount of low cost wireless sensor nodes can be densely deployed in the environment of interest. Usually, such kind of nodes have three major functions: sensing, processing and communications. The nodes sense the environment and communicate the result to the user.

In this paper, we use Xbow wireless sensor network professional developer's kit MOTE-Kit[3] as our testbed to get several data sets, from different scenarios. First of all, we show that the sensed signal strength is self-similar and long-range dependent using *variance-time plotting* , a common statistical method which has been widely used to verify self-similarity of time-series. Since the sensed signal strength is self-similar, its characteristics can be captured. We apply a type-1 FLS and an interval type-2 FLS to do sensed signal strength forecasting. This study can be extended to power on/off control in wireless sensors to save energy consumption, which is one of the key issues of wireless sensor networks.

In the following sections, Section II studied the self-similarity of sensed signal strength; the sensed signal strength forecasting using type-1 FLS and type-2 FLS are presented in Section III; and conclusions and future works are provided in Section IV.

II. Self-Similarity of Sensor Network Data

For a detailed discussion on self-similarity in time-series, see [12] [11]. Here we briefly present its definition [1].Given a zero-mean, stationary time-series $X = (X_t; t = 1, 2, 3, \cdots)$, we define the m-aggregated series $X^{(m)} = (X_k^{(m)}; k = 1, 2, 3, \cdots)$ by summing the original series X over nonoverlapping blocks of size m. Then it's said that X is H-self-similar, if, for all positive m, $X^{(m)}$ has the same distribution as X rescaled by m^H. That is,

$$X_t \triangleq m^{-H} \sum_{i=(t-1)m+1}^{tm} X_i \quad \forall m \in N \qquad (1)$$

If X is H-self-similar, it has the same autocorrelation function $r(k) = E[(X_t - \mu)(X_{t+k} - \mu)]/\sigma^2$ as the series $X^{(m)}$ for all m, which means that the series is distributionally self-similar: the distribution of the aggregated series is the same as that of the original.

Self-similar processes can show *long-range dependence*. A process with long-range dependence has an autocorrelation function $r(k) \sim k^{-\beta}$ as $k \rightarrow \infty$, where $0 < \beta < 1$. The degree of self-similarity can be expressed using *Hurst* parameter $H = 1 - \beta/2$. For self-similar series with long-range dependence, $1/2 < H < 1$. As $H \rightarrow 1$, the degree of both self-similarity and long-range depence increases.

One method that has been widely used to verify self-similarity is the *variance-time plot*, which relies on the slowly decaying variance of a self-similar series. The variance of $X^{(m)}$ is plotted against m on a log-log plot, and a straight line with slope $(-\beta)$ greater than -1 is indicative of self-similarity, and the parameter H is given by $H = 1 - \beta/2$. We use this method to verify the self-similarity of acoustic signal.

In our experiments, 8 sensors were deployed in a lab. The location of the sensors is showed in Fig. 1. We designed two scenarios, one is with a fixed source, and the other is without. In Fig. 2, we plot the variance of $X^{(m)}$ against m on a log-log plot for 8 sensor nodes respectively in the first scenario and Fig. 3 is under the second scenario. From the two figures, it's very clear that the no matter under what kind of condition the sensor network data have self-similarity because their traces have slopes much greater than -1.

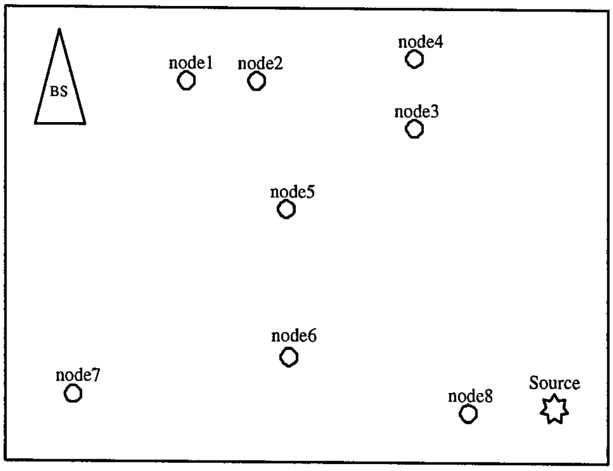


Fig. 1. The deployment of the eight sensor nodes in our experiments.

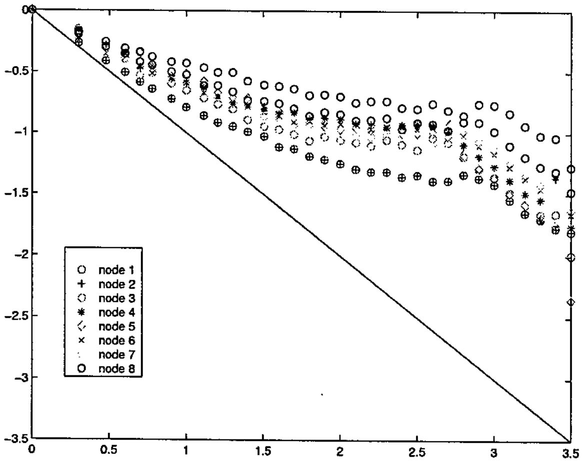


Fig. 2. The *variance-time* plot for sensed signal strength without music as background during 3 hours. The sampling rate is 1024ms/sample.

III. Sensed Signal Strength Forecasting using interval type-2 FLS

Since the sensed signal strength is self-similar, its characteristics can be captured. We applied a type-1 and an interval type-2 FLS to sensed signal strength forecasting. Both the two FLSs have 16 rules for sensed signal strength modelling. The parameters associated with each rule are determined based on the first 500 sensor data, and the steepest descent algorithm is used to optimize the parameters. We used the tuned type-1 and interval type-2 FLS to forecast the sensed signal strength following the training data. 500 sensor data are used for testing, and the forecasting performances are evaluated in terms of root-mean square error.

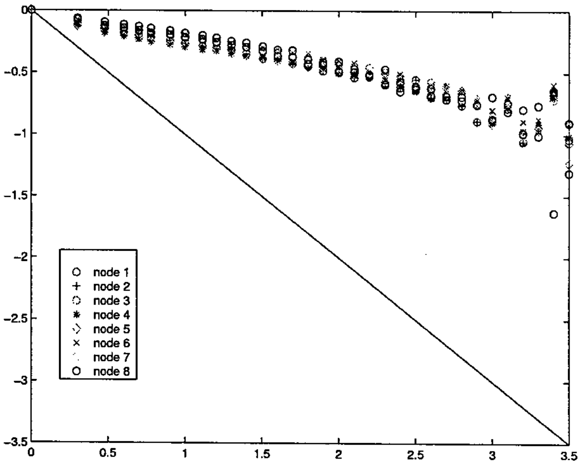


Fig. 3. The *variance-time* plot for sensed signal strength without music as background during 3 hours. The sampling rate is 1024ms/sample.

A. Introduction to Type-2 Fuzzy Sets

The concept of type-2 fuzzy sets was introduced by Zadeh [13] as an extension of the concept of an ordinary fuzzy set, i.e., a type-1 fuzzy set. Type-2 fuzzy sets have grades of membership that are themselves fuzzy [2]. A type-2 membership grade can be any subset in $[0,1]$ – the *primary membership*; and, corresponding to each primary membership, there is a *secondary membership* (which can also be in $[0,1]$) that defines the possibilities for the primary membership. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity. Type-2 fuzzy sets allow us to handle linguistic uncertainties, as typified by the adage "words can mean different things to different people." A fuzzy relation of higher type (e.g., type-2) has been regarded as one way to increase the fuzziness of a relation, and, according to Hisdal, "increased fuzziness in a description means increased ability to handle inexact information in a logically correct manner [5]".

Figure 4 shows an example of a type-2 set. The domain of the membership grade corresponding to $x = 4$ is also shown. The membership grade for every point is a Gaussian type-1 set contained in $[0,1]$, such a set was called a "Gaussian type-2 set". When the membership grade for every point is a crisp set, the domain of which is an interval contained in $[0, 1]$, Such type-2 sets were called "interval type-2 sets" and their membership grades "interval type-1 sets". Interval type-2 sets are very useful when we have no other knowledge about secondary memberships.

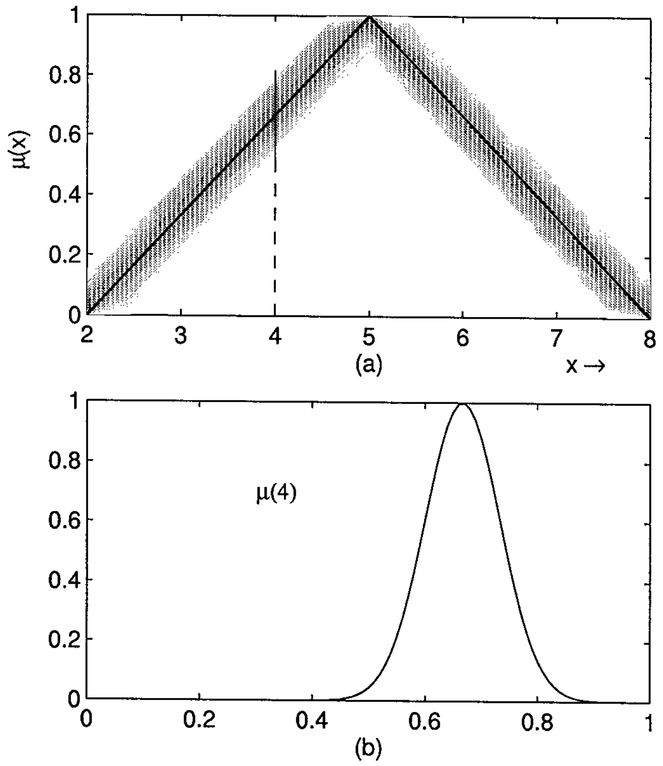


Fig. 4. (a) Pictorial representation of a Gaussian type-2 set. The secondary memberships in this type-1 fuzzy set are shown in (b), and are Gaussian. Note that this set is called a Gaussian type-2 set because all its secondary membership functions are Gaussian. The "principal" membership function (the bold line), which is triangular in this case, can be of any shape.

B. Introduction to Type-2 Fuzzy Logic Systems: An Overview

Figure 5 shows the structure of a type-2 FLS. It is very similar to the structure of a type-1 FLS [9]. For a type-1 FLS, the *output processing* block only contains the defuzzifier. We assume that the reader is familiar with type-1 FLSs, so that here we focus only on the similarities and differences between the two FLSs.

The fuzzifier maps the crisp input into a fuzzy set. This fuzzy set can, in general, be a type-2 set.

In the type-1 case, we generally have "IF-THEN" rules, where the lth rule has the form "R^l : IF x_1 is F_1^l and x_2 is F_2^l and $\cdots$ and x_p is F_p^l, THEN y is G^l", where: x_is are inputs; F_i^ls are antecedent sets ($i = 1, \ldots, p$); y is the output; and G^ls are consequent sets. The distinction between type-1 and type-2 is associated with the nature of the membership functions, which is not important while forming rules; hence, the structure of the rules remains exactly the same in the type-2 case, the only difference being that now some or all of the sets involved are of type-2; so, the lth rule in a type-2 FLS has the form "R^l : IF x_1 is $\tilde{\mathrm{F}}_1^l$ and x_2 is $\tilde{\mathrm{F}}_2^l$ and $\cdots$ and x_p is $\tilde{\mathrm{F}}_p^l$, THEN y is $\tilde{\mathrm{G}}^l$".

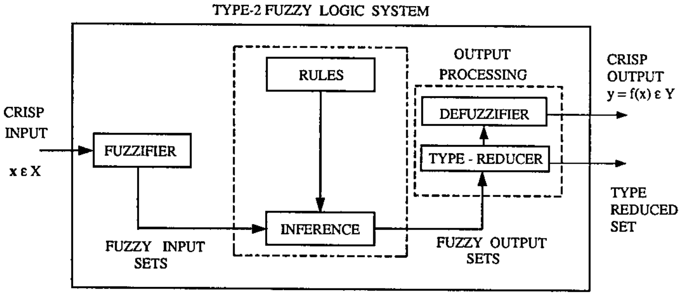


Fig. 5. The structure of a type-2 FLS. In order to emphasize the importance of the type-reduced set, we have shown two outputs for the type-2 FLS, the type-reduced set and the crisp defuzzified value.

In the type-2 case, the inference process is very similar to that in type-1. The inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. To do this, one needs to find unions and intersections of type-2 sets, as well as compositions of type-2 relations.

In a type-1 FLS, the defuzzifier produces a crisp output from the fuzzy set that is the output of the inference engine, i.e., a type-0 (crisp) output is obtained from a type-1 set. In the type-2 case, the output of the inference engine is a type-2 set; so, "extended versions" (using Zadeh's Extension Principle [13]) of type-1 defuzzification methods were used. This extended defuzzification gives a type-1 fuzzy set. Since this operation takes us from the type-2 output sets of the FLS to a type-1 set, this operation was called "type-reduction" and the type-reduced set so obtained a "type-reduced set".

To obtain a crisp output from a type-2 FLS, we can defuzzify the type-reduced set. The most natural way of doing this seems to be by finding the centroid of the type-reduced set; however, there exist other possibilities like choosing the highest membership point in the type-reduced set.

General type-2 FLSs are computationally intensive, because type-reduction is very intensive. Things simplify a lot when secondary membership functions (MFs) are interval sets (in this case, the secondary memberships are either 0 or 1). When the secondary MFs are interval sets, the type-2 FLSs were called "interval type-2 FLSs". In [7], Liang and Mendel proposed the theory and design of interval type-2 fuzzy logic systems (FLSs). They proposed an efficient and simplified method to compute the input and antecedent operations for interval type-2 FLSs, one that is based on a general inference formula for them. They introduced the concept of upper and lower membership functions (MFs) and illustrated the efficient inference method for the case of Gaussian primary MFs. They also proposed a method for designing an interval type-2 FLS in which we tune its parameters.

C. Why Type-2 FLS is Necessary?

FLSs have been extensively used in time-series forecasting (e.g., [4], [10], [7]). Here we designed a sensed signal strength forecasting scheme for wireless sensors using interval type-2 FLS. Why choose type-2 FLS? Acoustic amplitude sensor node measures sound amplitude at the microphone. Assuming that the sound source is a point source and sound propagation is lossless and isotropic, a root-mean-squared (RMS) amplitude measurement z is related to the sound source position X as

$$z = \frac{a}{\|X - \varsigma\|} + w, \tag{2}$$

where a is the RMS amplitude of the sound source, ς is the location of the sensor, and w is RMS measurement noise [6]. w is modelled as a Gaussian with zero mean and variance σ^2. The sound source amplitude a is also modelled as a random quantity, which is uniformly distributed in the interval $[a_{lo}, a_{hi}]$. Given the location of the sound source X and the sensor position ς, $\frac{a}{\|X-\varsigma\|}$ is uniformly distributed as a is. Therefore, z should be modelled as a Gaussian primary MF having a fixed standard deviation, σ_k^l, and an uncertain mean that takes on values in $[m_{k1}^l, m_{k2}^l]$,i.e.,

$$\mu_k^l(x_k) = \exp\left[-\frac{1}{2}(\frac{x_k - m_k^l}{\sigma_k^l})^2\right], \quad m_k^l \in [m_{k1}^l, m_{k2}^l] \tag{3}$$

where: $k = 1, \ldots, p$; p is the number of antecedents; $l = 1, \ldots, M$; and, M is the number of rules. It is shown in Fig 6.

D. Simulations

Our simulations were based on $N = 1000$ samples, $x(1)$, $x(2)$, ..., $x(1000)$. The first 500 data, $x(1)$, $x(2)$, ..., $x(500)$, are for training, and the remaining 500 data, $x(501), x(502), \ldots, x(1000)$ are for testing. In Fig. 7, we plot the sensed data that we used for training and testing, $x(1)$, $x(2)$, ..., $x(1000)$.

We used four antecedents for forecasting, i.e., $x(k-3)$, $x(k-2)$, $x(k-1)$, and $x(k)$ were used to predict $x(k+1)$. For type-1 FLS, the rules are designed such as:

R^l : IF $x(k-3)$ is F_1^l and $x(k-2)$ is F_2^l and $x(k-1)$ is F_1^l and $x(k)$ is F_2^l, THEN $x(k+1)$ is G^l.

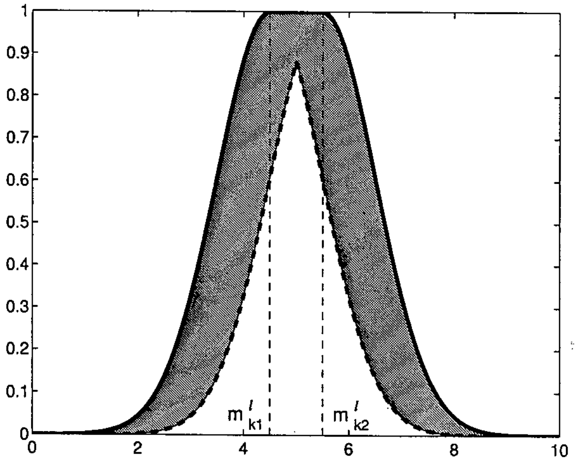


Fig. 6. The interval type-2 MFs with fixed std and uncertain mean.

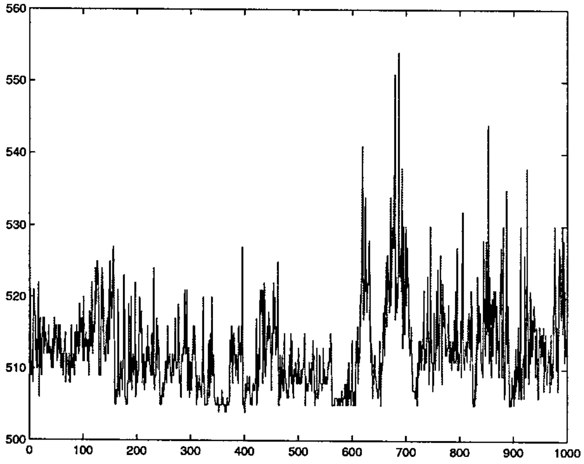


Fig. 7. Sensed data for 1024 seconds. the sample rate is 1024ms/sample, $x(1)$, $x(2)$, ..., $x(1000)$.

For interval type-2 FLS, we design the rules as:

R^l : IF $x(k-3)$ is $\tilde{\mathrm{F}}_1^l$ and $x(k-2)$ is $\tilde{\mathrm{F}}_2^l$ and $x(k-1)$ is $\tilde{\mathrm{F}}_1^l$ and $x(k)$ is $\tilde{\mathrm{F}}_2^l$, THEN $x(k+1)$ is $\tilde{G}^l$.

In this paper, we use height defuzzifier [9] for type-1 FLS, the height of G^l is $\overline{y}^i$. For interval type-2 FLS, we use Center-of-Sets type reduction [7].

As in [7], we used only 2 fuzzy sets for each antecedent, so the number of rules is $2^4 = 16$. For type-1 FLS, Gaussian membership functions (MFs) were chosen for the antecedents; for interval type-2 FLS, Gaussian primary MFs with fixed std and uncertain mean were chosen for the antecedents. The initial locations of antecedent MFs were based on the mean, m_t, and std, σ_t, of the first 500 points, $x(1)$, $x(2)$, ..., $x(500)$.

We used steepest descent algorithm to train all the parameters based on the training data. After training, the rules were fixed, and we tested the FL forecaster based on the remaining 500 noisy points, $x(501)$, $x(502)$, ..., $x(1000)$.

We compared the performance of the interval type-2 FLS with the type-1 FLS for sensed signal strength forecasting. For each of the 2 above methods, we ran 100 Monte-Carlo realizations to eliminate the randomness of the parameters, and the two FLSs was tuned using a simple steepest-descent algorithm. We used the testing data to see how each FLS performed by evaluating the root-mean-square-error (RMSE) between the defuzzified output of the FLS and the actual sensor data ($x(k+1)$), i.e.,

$$\text{RMSE} = \sqrt{\frac{1}{496}\sum_{k=504}^{999}[x(k+1) - f(\mathbf{x}^k)]^2} \qquad (4)$$

where $\mathbf{x}^k = [x(k-3), x(k-2), x(k-1), x(k)]^T$, and T denotes transpose.

The RMSE values for each design, we summarize the mean of the RMSEs and shown in Table I.

TABLE I

RMSE OF TYPE-1 FLS AND INTERVAL TYPE-2 FLS FOR SENSED SIGNAL STRENGTH FORECASTING.

System	RMSE
type-1 FLS	9.41
interval type-2 FLS	3.85

The results show that the interval type-2 FLS for sensed signal strength forecasting is much better than using type-1 FLS.

IV. Conclusions and Future Works

In this paper, we studied the sensed signal strength using the data collected in MOTE-Kit[3] testbed and show that the sensor data are self-similar, which validate that sensed signal strength is forecastable because self-similar time-series can be forecasted. Based on the analysis of the sensor model, we applied an interval type-2 FLS to sensed signal strength forecasting, and simulation results show that it performs much better than does a type-1 FLS. All these studies are very important for power on/off control in wireless sensors to save battery energy, which are the future works that we will investigate.

References

[1] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE Trans. on Networking*, vol. 5, no. 6, pp. 835-846, Dec 1997.

[2] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, USA, 1980.

[3] J. L. Hill, and D. E. Culler, " Mica: A Wireless Platform for Deeply Embedded Networks ," *IEEE, Micro*, Volume: 22, Issue: 6, pp. 12-24, Nov.-Dec. 2002.

[4] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, May/June 1993.

[5] E. Hisdal, "The IF THEN ELSE statement and interval-valued fuzzy sets of higher type," *Int'l. J. Man-Machine Studies*, vol. 15, pp. 385-455, 1981.

[6] L. E. Kinsler, A. R. Frey, A. B.Coppens and J. V. Sanders, *Fundamentals of Acoustic*, John Wiley and Sons, Inc., New York, USA, 1999.

[7] Q. Liang and J. M. Mendel, " Interval type-2 fuzzy logic systems: theory and design," *IEEE Trans. Fuzzy Systems*. vol. 8, no. 5, pp. 535-551, Oct 2000.

[8] E. H. Mamdani, " Applications of fuzzy logic to approximate reasoning using linguistic systems", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 26, no. 12, pp. 1182-1191, 1977.

[9] J. M. Mendel, "Fuzzy Logic Systems for Engineering : A Tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345-377, March 1995.

[10] J. M. Mendel and G. Mouzouris, " Designing fuzzy logic systems ," *IEEE Trans. Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 44, no. 11, pp. 885-895, Nov 1997.

[11] W. Stallings, *High-Speed Networks: TCP/IP and ATM Design Principles*, Upper Saddle River, NJ, 1998.

[12] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: statistical analysis of ethernet LAN traffic at the source level," *IEEE Trans. on Networking*, vol. 5, no. 1, pp. 71-86, Feb 1997.

[13] L. A. Zadeh, " The concept of a linguistic variable and its application to approximate reasoning - I," *Information Sciences*, vol. 8, pp. 199-249, 1975.

Energy and Delay Aware Packets Transmission in Wireless Sensor Networks

Xinsheng Xia
Department of Electrical Engineering
University of Texas at Arlington
416 Yates Street
Nedderman Hall, Rm 541
Arlington, TX 76010
Email: xia@wcn.uta.edu

Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
416 Yates Street
Nedderman Hall, Rm 541
Arlington, TX 76010
Email: liang@uta.edu

Abstract—**The performance evaluation is one the most important research topics for the Wireless Sensor Networks (WSN). Delay and energy efficiency are two important parameters to evaluate the WSN quality. In the WSN, the interference will affect the packets transmission. When a sensor need to send a packet, we choose the parameter SIR as the threshold to decide whether send or not. If the SIR threshold is high, the probability of the sending a packet will decrease and the packet need to be kept in the queue, and the delay of the packet will increase. If the SIR threshold is lower, the delay is decrease. However, in order to overcome the interference, the energy cost for each packet will increase. Simulation shows the SIR threshold can control the delay/energy performance in WSN.**

I. Introduction

Thanks to the rapid development in low power wireless communication, microprocessor hardware in conjunction with the significant process in distributed signal process, wireless sensor networks (WSNs) approach to a new technological vision. While a lot of research has been concentrated on the some important aspects of WSNs, such as energy efficiency, protocol design and network deployment, the performance evaluation in WSNs is rarely studied [1]. As the WSNs is widely deployed in military and commercial application [2], energy efficiency and delay-aware become more important for WSNs.

Interference is the major limiting factor in the performance of wireless sensor networks. There are several kinds of sources of interference. In this paper, we only consider the interference caused by neighboring sensors. When the neighboring sensors send out packets, they will affect the packet received by the destination sensor. In Figure 1, the source sensor is sending packets to the destination sensor, sensors B, C, E are also sending packets to their destination sensor. Becase every sensor has a non-directinal antenna, it will broadcast its packets to all the sensors around it. Sensors A, D are not sending packets at this time, they will not cause interference to the destination sensor. We calculate the interference in every small time period, so we can know the interference stays constant during each time period, but fluctuates randomly in different periods.

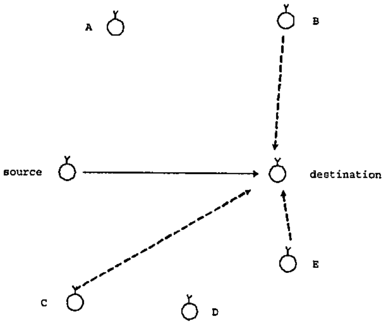


Fig. 1. the interference caused by neighboring sensors

The main dilemma that the transmitter faces is when the interference is very large, the packets the destination received will affected by the interference. So we can control the packet transmission as the following: when the transmitter observers high interference in the channel, that is the signal-to-interference ratio (SIR) less than a threshold, it would be better to back off, buffer the traffic and wait for the interference to subside before it transmits. As it has backed off, the buffer is filling up with new packet arrivals and delay rises.

In order to evaluate the algorithm, we need to understand the major parameters about the wireless sensor network.

A. Delay

Because data communications in the sensor networks has trimming constraints, it is important to design the network algorithm to meet a kind of end-end deadlines [3].

B. System lifetime

It is not convenient to recharge the sensor node battery, so the energy efficiency is extremely important for sensor network. And the network should keep an enough number of "live" sensor nodes to collect data, which means the network need to keep the energy among the sensor nodes in balance. We use the remaining alive sensor nodes as the parameter of the networks lifetime.

C. Network efficiency

The sensor network is used to collect data and transfer packets. The amount of packets transmitted is one of the parameters to evaluate the networks efficiency.

The rest of this paper is organized as follows. In section II, we introduce some preliminaries. In section III, we introduce the packet transmission scheduling. In section IV, we introduce the performance analysis. In section V, we introduce the simulation result. In section VI, we conclude the paper.

II. Preliminaries

A. Interference

Frequency reuse implies that in a given coverage area there is several wireless sensors that use the same set of frequencies. These sensors are called co-channel sensor, and the interference between signals from these sensors is called co-channel interference [4].

Let N be the number of co-channel interfering sensors, then the signal-to-interference ratio(SIR) for a mobile receiver which monitor a forwarding channel can be expressed as : ratio(SIR) for a mobile receiver which monitor a forwarding channel can be expressed as:

$$\frac{S}{I} = \frac{P_s}{P_I} = \frac{P_s}{\sum_{i=1}^{N} P_{I_i}} \tag{1}$$

Where P_s is the desired signal power from the desired sensor and P_{I_i} is the interference power caused by the ith interfering co-channel sensor.

B. Queueing Model

We setup a threshold to decide whether the source sensor send out a packet. The sensor is keeping collecting data with a Poisson distribution, while the service is general distribution.

The M/G/1 queue has exponentially distributed interarrival times, general process times, and a single server. The inclusion of general process times makes this formula more relevant for wafer fabs, where processing times are typically not as variable as the exponential distribution used with M/M/1 queues.

Let λ be the arrival rate, μ be the service rate and the $\rho = \frac{\lambda}{\mu}$ be the traffic, if we substitute in the variance plus the squared mean for the second moment of the service time distribution [5], we can get:

Average queuing length:

$$L_q = \frac{\lambda^2(Var[S_0] + \mu^{-2})}{2(1-\rho)} \tag{2}$$

Average queuing time:

$$W_q = \frac{L_q}{\lambda} = \frac{\lambda^{(}Var[S_0] + \mu^{-2})}{2(1-\rho)} \tag{3}$$

C. Energy

A sensor node consumes significant energy when it transmits or receives a packet. But we will not consider the energy consumed when the sensor node is idle.

The distance between two nodes are variable in the WSN and the power loss model is used. To send the packet, the sender consumes [6],

$$P_{tx} = P_{elec} + \epsilon_{fs} \cdot d^2 \tag{4}$$

and to receive the packet, the receiver consumes,

$$P_{rx} = P_{elec} \tag{5}$$

where P_{elec} represents the power that is necessary for digital processing, modulation, and ϵ_{fs} represents the power dissipated in the amplifier for the free space distance d transmission.

D. Delay

The packet transmission latency between the sensors includes three parts: the wireless channel transmission delay, the Physical/Mac layer transmission delay, and the queuing delay [7].

Defined D as the distance between two sensors and C as the light speed, the wireless channel transmission delay as:

$$Delay_{ch} = \frac{D}{C} \tag{6}$$

The Physical/Mac layer transmission delay will be decided by interaction of the transmitter and the receive channel, the sensor density and the sensor traffic intensity etc.

The queuing delay is decided by the sensor I/O system processing rate, the subqueue length in the sensor.

In order to make the system"stable", the rate at which sensor node transfers packets intended for its destination must satisfy all sensor that the queuing lengths will not be infinite and the average delays will be bounded.

III. Packet Transmission Scheduling

Time division multiple access(TDMA)system divided the radio spectrum into time slots, and in each slot only one user is allowed to either transmit or receive. The TDMA frame(Figure.2) has two segments: the header segment consists of request slots and the trailer segment consists of information slots. Request slots are minislots and an information slot is much longer in order to transfer data packets.

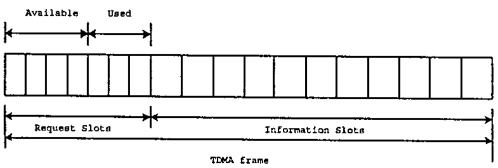


Fig. 2. TDMA frame

We setup a threshold to decide whether the source sensor send out a packet or insert the packet in its queue. The sensor is keeping collecting data with a Poisson distribution. When one packet is generated, insert the packet into the tail of the queue of the sensor, then compare the SIR of the destination sensor. If the SIR large than the threshold, the source sensor sends out its packet which locate at the head of the queue to the destination sensor, else keep idle.

A. SIR calculation

Every sensor calculates its SIR continuously. It needs to know the signal power S and interference power I_i caused by the ith interfering sensor. The transmission energy each packets cost can be decided by (4). It is a function of the distance between the source sensor and the destination sensor.

$$D(d,s) = \sqrt{(x_d - x_s)^2 + (y_d - y_s)^2} \tag{7}$$

Where x_d, y_d, x_s, y_s are X position, Y position of the destination sensor and the source sensor.

$$P_{I_i} = P_{s_i} - \varepsilon_{fs} * \frac{D(d,i_i)^2}{D(d,s)^2} \tag{8}$$

Where P_{s_i} is the power that the interference sensor used to send its own packets. P_{I_i} is the interference power generated by sensor I, ε_{fs} represents the power dissipated in the amplifier for the free space distance transmission.

$$SIR = \frac{P_S}{P_I} = \frac{P_{elec}}{\sum\limits_{i=1}^{N-1} P_{I_i}} \tag{9}$$

Where P_{elec} represents the power that is necessary for digital processing. According to (5), we konw is the power the receiver comsumes is P_{elec} .

According to the SIR calculation algorithm, the sensors need to exchange their information when they are sending packets. In the head of every packet, it will include the X position x_s, Y position y_s and the signal power P_s.

B. one-step Markov path model

The sensors are roaming independently with variable ground speed. The mobility model is called one-step Markov path model [8]. The probability of moving in the same direction as the previous move is higher than other directions in this model, which means this model has memory. Fig.3 shows the probability of the six directions.

IV. Performance Analysis

The delay of this WSN system can be analyzed using the M/D/1 queueing theory. However, we cannot achieve the variance of the general distribution. We consider approximate analysis of ON/OFF data traffic under light load assumptions that we believe most networks will operate in [9].

In the following, we assume the light load condition. To calculate the mean data packet queuing delay, let us first define some variables.

D_1 : denote the time interval from the time the data packet arrives in the ith TDMA frame to the beginning of the next TDMA frame.

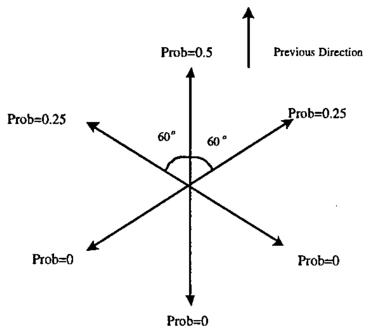


Fig. 3. one-step Markov path model

D_2 : denote the single TDMA frame that the data packets has to wait in the (i+1)th TDMA frame.

D_3 : denote the request slot subframe that the data packet has to wait in the (i+2)th TDMA frame.

D_4 : denote the time interval that the data packets have to wait for other data packets in the (i+2) frame.

The data packets queueing delay:

$$T_d = D_1 + D_2 + D_3 + D_4 \tag{10}$$

Let us assume the D_1 is uniformly distribution between the beginnings of two consecutive TDMA frames, e.g., between zero and T_{TDMA}, where T_{TDMA} is the TDMA frame interval. Furthermore, let us assume that there are m_d integer-multiple of one packet of size equivalent to one information slot packets generated at a constant interval of T_{TDMA}/m_d during the ON period in a TDMA frame interval, where m_d is the number of information slots used by a data source in a TDMA frame interval during its ON period. Its pdf is given by:

$$f_{D_1}(t) = \begin{cases} \frac{1}{T_{TDMA}}, if \ \ \frac{a-1}{m_d}T_{TDMA} \leq t < \frac{a}{m_d}T_{TDMA}, a = 1, 2, \ldots \\ 0, \ othewise \end{cases} \tag{11}$$

Where a is the arrival order of the packets in an TDMA frame time. Thus the mean of D_1 is:

$$E[D_1] = \frac{1}{2}T_{TDMA} \tag{12}$$

The pdf of D_2 is give by:

$$f_{D_2}(t) = \begin{cases} 1, if \ \ t = TDMA \\ 0, \ \ otherwide \end{cases} \tag{13}$$

while the pdf of D_3 is given by

$$f_{D_3}(t) = \begin{cases} 1, if \ \ t = N_R T_R \\ 0, \ \ otherwise \end{cases} \tag{14}$$

Where N_R is the number of request slots in the request slot subframe and T_R is the time duration of one request slot. That is, the means of D_2 and D_3 are respectively given by:

$$E[D_2] = T_{TDMA}, \tag{15}$$

and

$$E[D_3] = N_R T_R \tag{16}$$

Let N_d be the number of data users in the active (ON) state that constitutes a self-similar traffic. The pdf of N_d is given by:

$$P_r[N_d = n_d] \approx \left(\begin{array}{c} M_d \\ n_d \end{array} \right) P_{on,d}^{n_d}(1 - P_{on,d})^{M_d - n_d} \tag{17}$$

Where $P_{on,d} \approx \frac{t_1}{t_1+t_2}$, t_1 is the mean ON period for a ON/OFF data traffic and t_2 is the mean OFF period for the ON/OFF data traffic. The conditional probability of N_d is given by:

$$P_r[N_d = n_d | N_d > 0] = P_r[N_d = n_d]/(1 - P_r[N_d = 0]) \tag{18}$$

Let S be the total number of data packets generated in a TDMA frame interval. Its pdf is given by:

$$P_r[S = s = m_d N_d] = P_r[N_d = n_d = S/m_d] \tag{19}$$

Each of $k \times m_d$ data packets is assumed to have equal probability of transmitting in each of the $k \times m_d$ consecutive information slots. The pdf of D_4 given a, where a=1,2 ,3,...m_d, is given by:

$$\begin{aligned} & f_{D_4|a}(t = ((i-1)m_d + a - 1)T_I) \\ & = \frac{1}{m_d} \sum_{k=i}^{M_d} \frac{1}{k} P_r[N_d = k | N_d > 0] \delta[t - ((i-1)m_d + a - 1)T_I] \\ & i = 1, 2, \cdots, M_d \end{aligned} \tag{20}$$

Where $\delta(\cdot)$ is an impulse function. Its mean is given by

$$\begin{aligned} E[D_4] = & \sum_{a=1}^{m_d} \sum_{i=1}^{M_d} \frac{1}{m_d} \sum_{k=i}^{M_d} \\ & \frac{1}{k} P_r[N_d = k | N_d > 0]((i-1)m_d + a - 1)T_I \end{aligned} \tag{21}$$

V. Simulation

We implemented the simulation model using the OPNET modeler. The simulation region is 800×800 meters. In order to overcome the interference, the sensor would cost different energy with different SIR threshold. We chose the SIR thresholds as three cases: high(0.5), medium(0.3) and low(0.1).

There were 12 sensor nodes in the simulation model, and the sensors were roaming independently with variable ground speed between 1 to 9 meters per second. The mobility model was called one-step Markov path model. The movement would change the distance between sensor and it would also change the interfernce power.

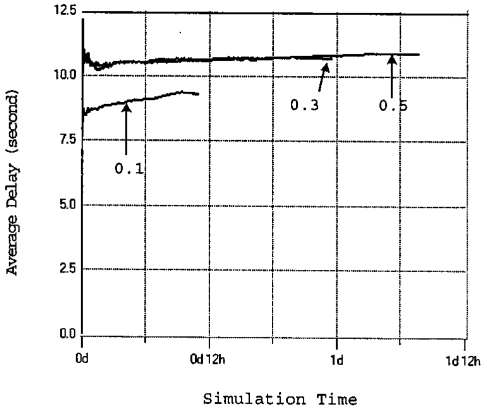


Fig. 4. Average Delay performance of the three thresholds

1) Average Delay: We used the average delay parameter [see (22)] to evaluate the network performance. Each packet was labeled a timestamp when it was generated by the source sensor node. When its destination sensor node received it, the time interval was the transmission delay.

$$Average\,Delay = \frac{\sum_{i=1}^{K} D_i}{K} \tag{22}$$

Observing Fig.4 the delay performance of three thresholds, the lower threshold, the better delay performance. When we chose the low SIR threshold, the sensor had a high probability to send out a packet than high and medium cases. The average delay of the low threshod around 21% was shorter than that of the high case. However, the average delays of high and medium case were almost the same. The reason was that in most cases, the interference was between 0.1 and 0.3. If the service is time-sensitive, such as video or audio service, we would choose the low SIR threshold to meet their delay performance requirement.

2) Energy Efficiency: When we chose the lower SIR threshold, that meant we need to increase the transmission power to overcome the interference. According to (4) and (5), for high SIR case, the sensor node consumed P_{tx} watts during transmissions and P_{rx} watts during reception. We assumed P_{elec} was equal to 6.0×10^{-4} and ϵ_{fs} was equal to 6.0×10^{-4}. For medium SIR case, both the transmission and reception energy comsumed would be 1.67 times of those of the high SIR case. And for low SIR threshold case, both the transmission and reception energy comsumed would be 5 times of those of the high SIR case. The lower SIR threshold, more energy comsumed. In the wireless sensor network, we used two parameters: the number of sensor nodes alive and the remaining energy to describe the energy efficiency.

When the remaining energy of a sensor node was lower than a certain threshold, the sensor was considered as "dead". In this simualtion, we chose 1.2×10^{-3} as the threshold. A sensor was "dead" meant it could not transmit/receive packets

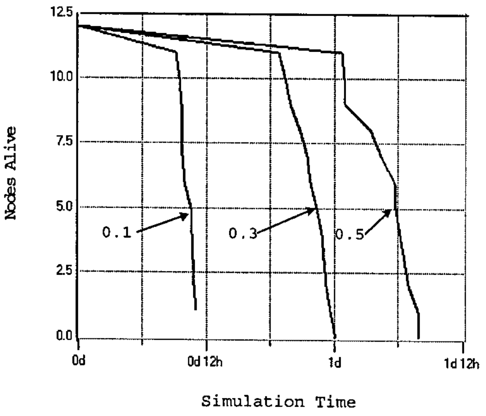


Fig. 5. Node alive performance of the three thresholds

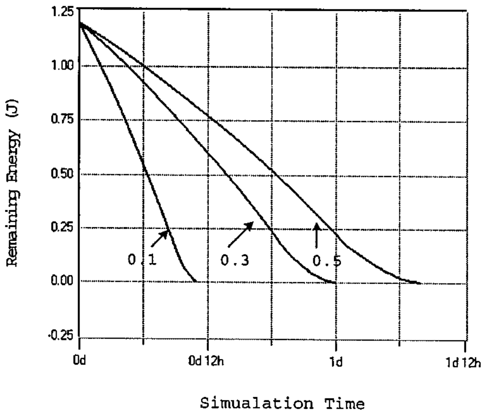


Fig. 6. Remaining energy of the three thresholds

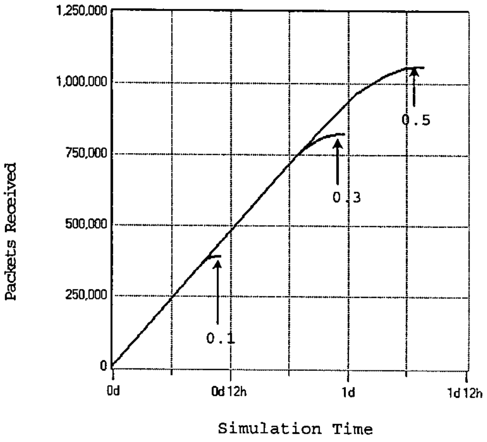


Fig. 7. Packets received of the three threshold

any longer, so it would be ignored by the sensor network. The sensor was used to collect data and transmit the packets. The number of sensor of a wireless sensor networks which was below a certain threshold means this network does not work.

As Fig.5 showed, the remaining sensor nodes alive of the lowe SIR threshold was much worse that of the high case and the medium case was between the low case and high case. The reason was, the higher SIR threshold, less energy comsumed. The time of the first sensor node "dead" of the high SIR threshold was over two time longer of that of the low SIR threshold case.

Fig.6 showed the remaining energy of the three thresholds. We assumed that the energy of each sensor was 0.1J and the packet size was 600 bit, and the channel transmission rate was 1M bps. So when the sensor transmitted or received a packet, it would cost 6×10^{-4} second. If a sensor node transmitted Num_s packets (each packet cost 6×10^{-4} second) and receives Num_r packets (each packets also cost 6×10^{-4} second) and it was roaming in the network for T_m, we could get the remaining energy E_i of this sensor node:

$$E_i = 0.1-(P_{tx}\times 6\times 10^{-4}\times Num_s+P_{rx}\times 6\times 10^{-4}\times Num_r) \tag{23}$$

The remaining energy E_w of the whole networks was described as:

$$E_w = \sum_{i=1}^{12} E_i \tag{24}$$

Figure.7 showed the remaining energy of the high SIR threshold was not dropping as sharply as that of the the low SIR threshold. The medium was between them.

3) Network quality: The role of the wireless sensor network in the real world was to collect data and transmit packets. In our simulation, we assumed the collecting data distribution of the sensor node was Poisson distribution and the arriving interval was 1 second. Observing from Fig.7, the higher SIR threshold, more packets collected. It was because the energy cost by each packet in the high case is less that of the low case.

VI. Conclusion

In the wireless sensor network, the interference will greatly affect the packets transmission. If the SIR is high, the probability of the sending a packet will decrease and the packet need to be kept in the queue. The delay of the packet will increase. If the SIR is lower, the delay is decrease. However, in order to overcome the interference, the energy cost for each packet will increase. We realize the wireless communcation system with TDMA and we analyze the performance of delay and energy according to different SIR threshold. Simulation shows the SIR threshold can control the delay/energy performance in WSN.

Acknowledgment

This work was supported by the U.S. Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466.

References

[1] Chen, D. et al; " QoS Support in Wireless sensor Networks: A Survey ," *International MultiConference in Computer Science and Computer Engineering,ICWN'04*, Las Vegas, Nevada, USA, June 21-24, 2004

[2] Akyildiz, I. et al; " A survey on sensor networks ," *IEEE Coummun. Magazine*, pp.102-114, Aug. 2002

[3] Lu, C. et al; " RAP: a real-time communication architecture for large-scale wireless sensor networks ," *Proceeding of the eighth IEEE real-time and embedded technology and applications Symposium*, September 25 - 27, 2002,San Jose, California, Pages:55 - 66

[4] Rappaport, T.; " Wireless Communications: principles and practice ," *Second Edition*, Pearson Eduation, Inc., Pages: 68 - 69,

[5] Prabhu, N.; " Basic Queueing Theory. Technical Report No. 478 ," School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York.,

[6] Heinzelman, W.B.; Chandrakasan, A.P.; Balakrishnan, H.; " An application-specific protocol architecture for wireless microsensor networks ," *IEEE Transactions on Wireless Communications*, Volume: 1 Issue: 4, Oct 2002

[7] Xia, X.;Liang, Q.; "Latency-aware and energy efficiency tradeoffs for sensor networks" *Accepted by Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE* ,

[8] Hou T. C.;Tsai T. J.; "Adaptive clustering in a hierarchical ad hoc network" *Proc. Int. Computer Symp.,Tainan, Taiwan,R.O.C.*, Dec.1998, pp.171-176.,

[9] Wong, T.C.et al; " Delay performance of data traffic in a cellular wireless ATM Network ," *International Workshop on Wireless Mobile Multimedia, WOWMOM 2000*, Proceedings of the 3rd ACM international workshop on Wireless mobile multimedia, Boston, Massachusetts, United States, Pages: 101 - 107

An Asynchronous Contention-Based and Energy-Efficient MAC Protocol (ACEMAC) for Wireless Sensor Networks

Qingchun Ren
Department of Electrical Engineering
University of Texas at Arlington
416 Yates Street
Nedderman Hall, Rm 205
Arlington, TX 76019
Email: ren@wcn.uta.edu

Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
416 Yates Street
Nedderman Hall, Rm 518
Arlington, TX 76019
Email: liang@uta.edu

***Abstract*—Wireless Sensor Networks (WSNs) pose considerable technical challenges in energy efficiency for resource constrained (energy, bandwidth, computation and memory). In this paper, we propose an asynchronous contention-based and energy-efficient MAC protocol (ACEMAC). Our algorithm ameliorates the biggest limitation for existed contention-based MAC protocols, time synchronization needed. We propose a designing model to optimum power off duration, power on duration and system stage construct. We also make theory analysis on average latency (Point-to-Point) for our algorithm. Simulation results verify that our algorithm can successfully obtain the optimum reschedule duty-cycle, power off duration and power on duration to achieve higher average successful transmission rate, shorter data packet average latency and longer network lifetime.**

I. Introduction

Wireless sensor networking is an emerging technology that has a wide range of potential applications including: military sensing, physical security, air traffic control, traffic surveillance, video surveillance, industrial and manufacturing automation, distributed robotics, environment monitoring, and structures monitoring[15]. However, individual sensor nodes are resource constrained (energy, bandwidth, computation, memory). So WSNs pose considerable technical challenges in energy efficiency, spectral efficiency and security. In this paper, we focus on the energy efficiency problem for WSNs.

There are two approaches for energy reservation on communication. One approach does power control at physical layer. The other approach implements energy reservation task on MAC layer. Considerable energy in traditional MAC protocol is used by idle listening, i.e., listening to receive possible traffic that is not sent. For example, the digital 2 Mbps Wireless LAN module (IEEE 802.11/2Mbps[10]) specification shows the energy consuming ratio of idle: receive: send is1:2:2.5[14]. So about one fifth of total energy, which should be saved, is consumed. It is obvious that enforcing some sensor nodes, which are staying at idle state, to turn off their RF interfaces is an effective method to implement energy reservation

In literature, some energy-efficient MAC protocols for WSNs are proposed. The first energy-efficient MAC protocol for ad hoc networks: PAMAS[5] is proposed in 1999. PAMAS conserves battery power by intelligently powering off users that are not actively transmitting or receiving packets. But the disadvantage for this protocol is, two different physical channels are needed, i.e., control channel and traffic channel. After PAMAS, some solutions have been proposed for WSNs. They can be classified into two categories: Schedule-based and Contention-based. TRAMA[2], a schedule-based MAC protocol, employs a traffic adaptive and distributed election scheme to allocate the system time for different sensor nodes. According to the election scheme, receivers are selected basing on schedules announced by transmitters. For S-MAC[3], a contention-based MAC protocol, borrows the main idea from PAMAS to implement energy reservation. But S-MAC has only one common channel to transmit data and control information. Considering the fixed duty-cycle is not optimal, S-MAC's improvement version T-MAC is proposed in [4]. For S-MAC and T-MAC, they tightly depend on time synchronization. Accurate time synchronization method is the premise to guarantee all nodes switch between sleep state and active state simultaneously.

In this paper, we propose a novel contention-based MAC protocol: asynchronous contention-based and energy-efficient MAC protocol (ACEMAC). Our algorithm overcomes the disadvantage for S-MAC and T-MAC, tightly depending on time synchronization[6], [7], [8], [9]. ACEMAC sets up system time through free-run timing method. At the same time, clock drift aware method is used to solve schedule un-coincidence of different sensor nodes. And no extra control channel is needed. We use *node* to stand for sensor node in the following parts.

The remainder of this paper is organized as follows. In section II, we provide some assumptions and the network model of our algorithm. Our algorithm is described in section III. Average latency is analyzed in section IV. Simulation results are given in Section V.And Section VI concludes this paper.

II. Assumptions and Network Model

We have following assumptions related to our algorithm:

- There are only two cases that can cause the packet transmission fail. One is collision between packets. The other is that the receiver is unavailable for receiving when transmitter sending packets;
- The errors caused by channel can be recovered by channel coding at physical layer;
- We can use certain clustering algorithm, such as LEACH[16], to cluster a whole WSNs into independent groups;
- The traffic for each node is the information detected and collected by this node;
- The information transmitted over the network is cluster-inside;
- For information transmission time, we ignore the process time at receivers and propagation time.

The network model that fits our algorithm is described as follows. For one cluster, there are some normal nodes and one cluster head. In the same cluster, there is only one hop between cluster head and normal nodes. If two normal nodes stay during each other's communication range, they are neighbors for each other. All normal nodes stay within cluster head's communication range. But for different normal nodes, they may not be neighbors with each other. Therefore, within one cluster, there can be several nodes making transmission at the same time.

III. Asynchronous Contention-Based and Energy-Efficient MAC Protocol (ACEMAC)

We have two goals when designing our algorithm. One is to implement energy reservation through scheduling all normal nodes powering off/on simultaneously. The other one is to obtain an optimal way to design the duration for power off/on. We exploit re-schedule method, a simple and easy implement way, to archive the coincidence among normal nodes instead of time synchronization mechanism. In order to resolve the problems caused by time variant environment, network topology and traffic strength of nodes, we design a clock drift aware system to dynamically and adaptively adjust the re-schedule duty-cycle. Our method belongs to contention-based MAC protocol category. In the following parts, we will discuss the detail about our algorithm.

A. System Time Scheme

Fig. 1 presents the system time scheme structure. ACEMAC divides system time into different stages. During different stages, node operates differently. The function for each stage is:

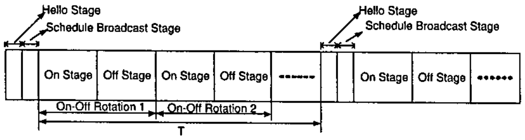


Fig. 1. System Time Scheme Structure

- "Hello Stage" is used by normal nodes to exchange HELLO message;
- "Schedule Broadcast Stage" is used by cluster head to broadcast schedules among cluster;
- "On Stage" is preserved for all nodes to power on communication interface. Therefore, transmitting and receiving can be implemented;
- "Off Stage" is preserved for all nodes to power off communication interface. Therefore, there is no transmitting and receiving happened. But data storage and sensing still continue;

From above figure, we can see that there are several rotations from power on to power off within the interval (T) between two continuous schedule broadcast.

B. System Schedule Set Up

ACEMAC schedules all nodes' transmissions into certain stages of system operation time. In other time, nodes reserve energy through powering off their RF interfaces. The premise for ACEMAC work smoothly is to set up a coincident system schedule correctly and effectively among nodes within on cluster.

Setting up a global clock within one group, as done for most TDMA system, is not a good choice for WSNs. The reasons are:

- Spectral efficiency requires the overhead brought by exchanging information among nodes for synchronizing the time of different nodes should be as little as possible;
- Energy efficiency brings forward cutting short the energy consumed by time synchronization processing;
- Limitation on computation and memory resources make it impossible to choose high quality, but complex time synchronization methods;

Our algorithm can overcome this defect of setting up global clock. We choose free-run timing method. That means, each node simply times itself to its own free-running local oscillator. In order to guarantee each node in the same cluster to switch to the same stage simultaneously basing on local clock, special schedule broadcast is used.

Cluster head makes a broadcast to inform the system schedule to every normal node in this cluster. In Fig. 3, schedule broadcast message format is shown. In this schedule broadcast message, $On-Duration$ field tells nodes the length of time after which they should end transmitting and receiving; $Off-Duration$ field tells nodes the length of time after which they should end current power on-off rotation.

Type	Source	On-Duration	Off-Duration

Fig. 2. Schedule Broadcast Packet Format

From above figure, we can see that a schedule broadcast packet regulates the duration of different stages, instead of the exact start time and end time. Even though each node uses its own local clock, basing on this schedule broadcast, all nodes could enter the same working state simultaneously according to our assumption.

Overcoming clock drift is another function for schedule broadcast, besides informing nodes system schedule. The quality of each local clock usually boils down to its frequency stability-that is, the ability of its frequency standard to emit events at a constant frequency over time. In general, as frequency standards' stability and accuracy increase, so do their power requirements, size and cost. For general WSNs, whose nodes are usually configured simply and inexpensively. Therefore, there are always some clock shifts caused by frequency standard inaccurate for nodes. Moreover, the frequency generated by a quartz oscillator is also affected by a number of environmental factors: the voltage applied to it, the ambient temperature, acceleration in space, magnetic fields, and so forth. More subtle effects as the oscillator ages also cause longer-term frequency changes. Under these conditions, after a period of time, system schedule un-coincident among nodes is unavoidable. See Fig. 3.

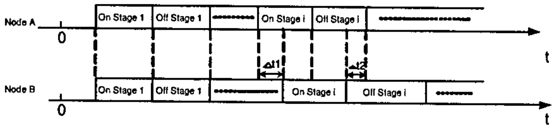


Fig. 3. Schedule Un-coincidence Due to Clock Drift

If the frequency standard for node A is faster than that of node B. From above figure, we can see that the transmitter, node A, precedes the receiver, node B, into the on stage. So during $\Delta t1$, data transmissions cannot be done successfully between node A and node B. We can solve this problem through doing re-schedule. See Fig. 4. After each re-schedule, $\Delta t1$ between node A and node B is removed.

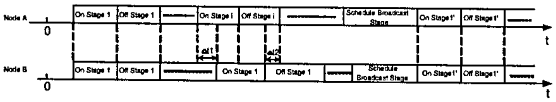


Fig. 4. Schedule Un-coincidence Removing Through Re-schedule

C. Re-schedule Duty-Cycle Design

From above discussion, we notice that keeping system schedule coincident among nodes can avoid transmission failures caused by receivers do not power on their RF interface. However, providing state switching simultaneously at all time will is not necessary. For example, there are two nodes. Their frequency standards have great discrepancy. So there is quite time difference between them. But there is seldom information exchange between them. This kind of system schedule un-coincidence almost has no any influence on information transmission. Therefore, some nodes can be allowed to go out of coincident, and then re-schedule only when needed. The aim of our algorithm is to reserve energy when ensuring certain successful transmission possibility.

For designing re-schedule duty-cycle (i.e., the length of the interval of two continuous schedule broadcast), we have two options. One is dynamically adjusting re-schedule duty-cycle method. The other is fixed re-schedule duty-cycle scheme.

It is obvious that for fixed duty-cycle, there is a large amount of energy used to send and receive the schedule broadcast, if the re-schedule duty-cycle is short. On the other hand, there is a great deal of energy wasted by fail information exchange, if the re-schedule duty-cycle is long. However, when most of fail transmissions are caused by clock drift, such energy for re-schedule is reasonable and worthy. So adaptively adjust method is a better choice.

We build a kind of clock drift aware scheme to dynamically adjust re-schedule broadcast duty-cycle. That is, when the cluster-head detects the clock difference among nodes influences the information transmission at very small degree, an acceptable one, it would prolong the duty-cycle. But once the cluster-head finds that the clock drift decreases the successful transmission rate sharply, it would shorten the duty-cycle. The details are discussed in the following part.

Each node would send out a re-schedule request after a fail transmission. In this paper, when the retransmission time exceeds certain value, we call it a fail transmission. It should be noted that retransmission is only for data packets. We use this message to inform the cluster head about the fail transmission ratio for this node. The format of re-schedule request packet is shown in Fig.5. We use the same model to determine the value of N_{ret} as in [13].

$$N_{ret} \approx 1.45 ln \frac{1}{1-p} \tag{1}$$

Where p is the probability of successful transmission.

Type	Source	Transmission Fail Rate

Fig. 5. Re-schedule Request Packet Format

According to the re-schedule requests received, cluster head adjusts the next re-schedule duty-cycle before doing re-schedule broadcast. That is

$$T_i = \begin{cases} 2 \times T_{i-1}\,, & when\ N\ decrease; \\ \frac{1}{2} \times T_{i-1}\,, & when\ N\ increase; \\ T_{i-1}\,, & otherwise. \end{cases} \tag{2}$$

where T_i is the next duty-cycle of re-schedule broadcast; T_{i-1} is the previous duty-cycle; and N is the number for cluster head shortening re-schedule duty-cycle. The range of N is [0, N_{max}]. If half number of nodes' fail transmission ratio are bigger than 5%, cluster-head during this period of time, it could add 1 to the last value of N. Otherwise, the cluster-head should reduce the last value of N by 1 until N=N_{max}.

D. HELLO Message Design

Hello stage is preserved for each node to broadcast its own HELLO message. Through this message, each node can inform neighbors that it is alive. Moreover, this message includes each node's traffic arrival rate (λ packets/second) and average service time(μ packets/second). This information is the considering factor for cluster-head to determine the length of On-stage and Off-stage. Fig. 2 shows the structure of a HELLO message. The traffic arrival rate and average service time are statistic values basing on previous traffic arrival rate and average service time. In our algorithm, we define a new terminology: coherent time. Coherent time is a special length of period, within which we assume that the environment condition, the channel condition, network topology and traffic strength are fixed. During every coherent time, we carry on measurement on traffic arrival rate and service time. In order to reduce the dependent degree on the cluster head, the estimations for traffic arrival rate and average service time are done by each normal node locally. Each node randomly chooses a time within the hello stage to send out HELLO message. We make this random process complies with uniform distribution. Transmission time's randomicity could reduce the collision probability for HELLO message and increase the successful transmission probability. In the following part, we will analyze the probability (P_s) for a normal node send HELLO message successfully.

Type	Source	Data Arrival Rate	Service Rate

Fig. 6. HELLO Message Format

$$P_s = 1 - \sum_{i=2}^{N} P\{n = i\} \tag{3}$$

where n is the number of nodes whose transmission is fail for collision. There are N normal nodes in this cluster.

$$P\{n = i\} = \left[\frac{\tau(2T_h - \tau)}{T_{ran}^2}\right]^i \times \left\lfloor \frac{i}{2} \right\rfloor \tag{4}$$

where τ is the transmission duration for a HELLO message. T_h denotes the duration of hello stage. Then the probability for all N nodes send message to cluster head successfully during HELLO state(T_h) is:

$$P_s = 1 - \sum_{i=2}^{N} \left[\frac{\tau(2T_h - \tau)}{T_h^2}\right]^i \times \left\lfloor \frac{i}{2} \right\rfloor \tag{5}$$

We can see that P_s is a function of parameter N and T_h.

E. Power On Duration(T_n) Design

For WSNs, according to the application field, there are lots of traffic types. We don't hope to limit our parameter design into one specific application. So basing on the traffic arrival rate, we design the sleep duration. If we know the average traffic, arrival rate is λ_i for node i. So the average number ($\bar{N}_i$) of packets arriving during the sleep part for node i is:

$$\bar{N}_i = \lambda_i T_{s,i} \tag{6}$$

According to (1), the value of $\bar{N}_i$ is proportional to $T_{s,i}$. We know that the buffer size for each sensor node is limited. When the buffer is used out, the following coming data packets must be discarded. We can obtain one criterion for determining the value of $T_{s,i}$: the number of arrival messages should not be bigger than the buffer size during the $T_{s,i}$ period. i.e.,

$$\bar{N}_i \leq K. \tag{7}$$

where K is the available buffer size. We assume that for each node, the buffer size is same. It is reasonable and realistic to suppose each node equipped with same resource.
Since $\bar{N}_i$=$\lambda_i T_{s,i}$. Then we have

$$\lambda_i T_{s,i} \leq K \rightarrow T_{s,i} \leq \frac{K}{\lambda_i}(i = 1, 2, \cdots, M) \tag{8}$$

We assume that there are M nodes in a cluster totally. The data arrival rate for every node of this cluster is identical and independent. So, for the same number of data packets arriving, the duration for different node might be different. We let the Cluster-head choose the shortest duration for the whole cluster. The sleeping duration for whole cluster is:

$$T_s = \min_i \left\{ \frac{K}{\lambda_i} \right\} \tag{9}$$

According to Little theorem[15], the average number(N_q) of customers in a queue system equals to the product of customer arrival rate(λ) and average time(ω) spent in the system:

$$N_q = \lambda w \tag{10}$$

For node i, we can get:

$$\bar{N}_i = \lambda_i w_i \tag{11}$$

Compare (5) and (10), we get

$$\lambda_i \left(T_{s,i} + \bar{T}_B\right) = \lambda_i w_i \rightarrow T_{s,i} + \bar{T}_B = w_i \tag{12}$$

From this equation, we get another criterion for choosing the value of $T_{s,i}$. For all sensor nodes, the permitted latency is W_{max}. So we can get:

$$T_{s,i} + \bar{T}_B \leq W_{max} \rightarrow T_s \leq W_{max} - \bar{T}_B \tag{13}$$

From (8) and (12), the optimum value of T_s should follow the rule:

$$T_s \leq \min\left\{ W_{max}, \min_i(\frac{K}{\lambda_i}) \right\} \tag{14}$$

In this paper, we set T_s as:

$$T_s = \min\left\{ W_{max}, \min_i(\frac{K}{\lambda_i}) \right\} \tag{15}$$

We know that, the longer the sleeping period is, the more energy reserved. However, the disadvantage is the average latency of data packets increasing. Energy is one of the most important resources for sensor networks. Reserving energy and reduce the latency are two contradict tasks.

F. Power Off Duration(T_f) Design

During active part, nodes start to send out data packets, which are received during last sleeping part and current active part. Through contention, each node obtains the right to hold the link. The contention progress is almost like CSAMA/CA[1]. Using carrier sense, RTS/CTS shake-hand and ACK confirmation, CSMA/CA ensures no collision for data packet transmission. And also, it permits the transmitter to know exactly that if the data packet transmission is successful or not. In our sleeping duration and active duration designing, we not only try to extend the sleeping time to reserve energy, but also ensure data packets' latency are not longer than acceptable value. So, the criteria for active duration is that it ($T_{a,i}$) is long enough for all received data packets to be sent out. Now, we should estimate the average number ($\bar{N}_{t,i}$) of data packets received during previous sleep part and current active part, then we can determine the value of $T_{a,i}$. Because the traffic arrival process is independent with the data transmission process, for $\bar{N}_{t,i}$ we have

$$\bar{T}_{t,i} = \bar{N}_{s,i} + \bar{N}_{a,i} = \lambda_i(T_s + T_a) \tag{16}$$

Basing on the $N_{t,i}$ data packets should be sent out during current active duration($T_{a,i}$), the following equation existed:

$$\mu_i T_{a,i} = \lambda_i(T_{a,i} + T_{s,i}) \tag{17}$$

The active duration for node i is:

$$T_{a,i} = \frac{\lambda_i T_s}{\mu_i - \lambda_i} \tag{18}$$

If the cluster-head chooses the longest duration as the whole cluster's active duration, for all nodes, each node could have enough time to send all received data packets out. Then we get the active duration for whole cluster. That is,

$$T_a = \max_i \left\{ \frac{\lambda_i T_s}{\mu_i - \lambda_i} \right\} \tag{19}$$

IV. System Performance Theory Analysis

Contention-based MAC protocol divides the system time into two types: sleep time and active time. For sleep time, each node can be regarded as a waiting queue. Data packets independently enter this queue at certain rate. But there is no any transmission. For active time, a group of nodes, locating in each other's communication range, could be treated as a M/M/1 queuing system. Within one cluster, all normal nodes stay within cluster-head's communication range. But for different normal nodes, maybe they are not neighbors. That depends on the topology of the cluster. So, the link could be successfully used by more than one transmitting-receiving pairs at one time. For this special characteristic, within one cluster, there are several M/M/1 queuing systems existed simultaneously. Looking through the system time, we cannot use M/M/1 queue model any more for introducing sleep time. We use M/G/1 Queue with vocation model[12] as our model. In this part, we analyze the average latency for one M/G/1 system. The average waiting time ($\bar{W}$) is:

$$\bar{W} = \frac{R}{1-\rho} \tag{20}$$

where R is the average residual time. ρ is the traffic intensity. There are

$$\rho = \frac{\lambda}{\mu} \tag{21}$$

and

$$R = \frac{\lambda \bar{X}^2}{2} + \frac{(1-\rho)\bar{V}^2}{2\bar{V}}. \tag{22}$$

where λ is the data arrival rate; μ is the average service time. $\bar{X}^2$ stands for second moment of service time, $\bar{X}$ stands for average service time and $\bar{V}$ stands for the average sleeping time. ρ is known if given λ and μ. In order to get the value of $\bar{W}$, we should compute $\bar{X}^2$. Since X=$\bar{T}_B(k)$+t_s. Then

$$\begin{aligned} \bar{X}^2 &= E[X^2] \\ &= E[\bar{T}_B^2(k)] + t_s^2 + 2t_s E[\bar{T}_B(k)] \end{aligned} \tag{23}$$

t_s is the average time that the link is sensed busy due to a successful transmission. Basing on [11], we have

$$\bar{T}_B\,(k) = \sum_{i=1}^{k} \frac{\alpha(W_i - 1)}{2} + (k-1)t_c \tag{24}$$

For $E[\bar{T}_B^2(k)]$, we have:

$$\begin{aligned} E[\bar{T}_B^2(k)] &= \sum_{k=1}^{\infty} (\bar{T_B})^2(k) P\{K = k\} \\ &= \sum_{k=1}^{\infty} [(\sum_{i=1}^{k} \frac{\alpha}{2}(w_i - 1))^2 + (k-1)^2(t_c)^2 \\ &\quad +2t_c(k-1)\sum_{i=1}^{k} \frac{\alpha}{2}(w_i - 1)](1-q)^{k-1}q \end{aligned} \tag{25}$$

We let

$$E\left[\bar{T}_B^2(k)\right] = \frac{\alpha}{2}(S_1 + S_2) \tag{26}$$

For S_1, we have:

$$S_1 = \sum_{k=1}^{m} [\sum_{i=1}^{k} (2^{i-1} w_{min} - 1)^2](1-q)^{k-1} q \tag{27}$$

For S_2, we have:

$$S_2 = \sum_{k=m+1}^{\infty} [(\sum_{i=1}^{m} (w_i - 1))^2 + (\sum_{i=m+1}^{k} (w_i - 1))^2](1-q)^{k-1} q \tag{28}$$

where m is the "maximum back-off stage". w_i is the minimum contention window size.

V. Simulations and Performance Evaluation

In this section we evaluate the performance of our method. OPNET is used to run simulators on the network. And the size of the network varies from 11 nodes to 22 nodes. All nodes transmit to some other node in the network according to the same source rate with fixed packet sizes. The communication range for each node is 20 meters. Nodes are randomly placed in an area of 100×100 meters and have no mobility. We trace each node in the network and compute the successful transmission rate, data packet average time delay and the time of first node dead. The experiment is repeated for 100 different seeds. We do that not only for statistical reasons, but also for the fairness problem inherent in the DCF of MAC protocol. Regarding the physical layer, we use Frequency-Hopping Spread Spectrum (FHSS) with a raw bit rate of 1Mbps. Other physical layer parameters are same as IEEE802.11 protocol[10].

We separately use our method and a derivate way, which only exchanges the sleeping part and active part of a frame. Under different clock shift rate, we compare the system successful transmission rate, average latency and the time of first node dead. From Fig. 7, we can see that with the clock shift rate increasing, there are more data packets transmission failed. When clock shift rate is larger than 0.005(i.e., there are 0.005 seconds drift every second), sleep-active method (i.e., for each frame, the active part follows sleeping part), the performances decrease sharply. The successful transmission rate decreases about 15.39%. But for active-sleep method, there is only 5.49% decrease. Our algorithm, active-sleep method, has better tolerance for clock shift. Fig. 8, shows that with the clock shift rate increases, the average time delay for data packet decreases. The reason is, we just consider the successful transmission. And a data packet would be discarded after several unsuccessful retransmissions. The average latency of the active-sleep method is almost 33.3% shorter than sleep-active method. From Fig. 9, we can see that the time of first node dead is brought forward with the clock shift rate increase. These simulation results also show that the performance of our system is not sensitive with the network density very much.

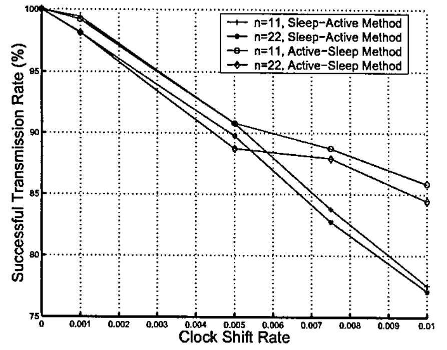


Fig. 7. Data Packet Successful Transmission Rate

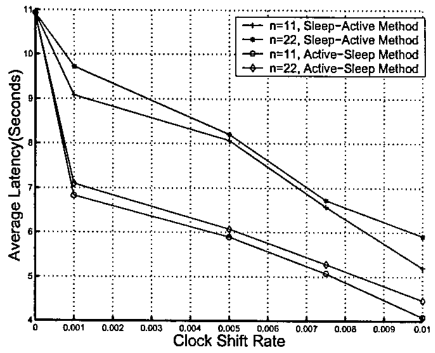


Fig. 8. Average Time Delay for Data Packet

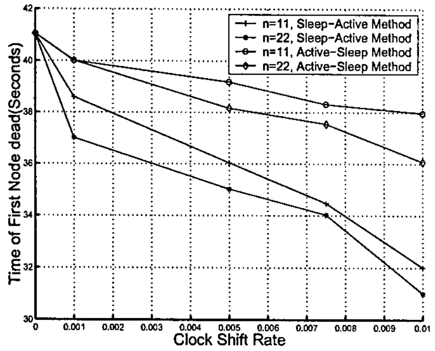


Fig. 9. Time of First Node Dead

Fig. 10, Fig. 11, Fig. 12 show the relationship between different sleeping duration and system performance. We use optimum, longer and shorter sleeping duration values separately. The sleeping duration, basing on our algorithm, let the successful transmission rate is almost same as all nodes always working without any sleep. Moreover, it prolongs the time of first node dead about 22.58%. We also can see that with the sleeping duration increase, the average latency increase.

We do the same experiments with the active duration. From Fig.13 and Fig. 15, we can get that, even the successful transmission rate for optimum active duration is a little lower than no sleep one, the time of the first node dead is prolonged about 22.23%. More energy are reserved.

In the last simulation, we do experiments with fixed duty-cycle and adaptively adjusted duty-cycle. There are two cases: one is the initial synchronization broadcast duty-cycle that is 2 times of frame time; the other one is the duty-cycle that is the 4 times of frame time. Under different clock shift rate, we compare the successful transmission rate.

VI. Conclusions

In this paper, we propose a novel energy-efficient MAC protocol: ACEMAC. Our algorithm is an asynchronous method, which completely overcomes the biggest disadvantage, i.e., so tightly depending on time synchronization, of existing

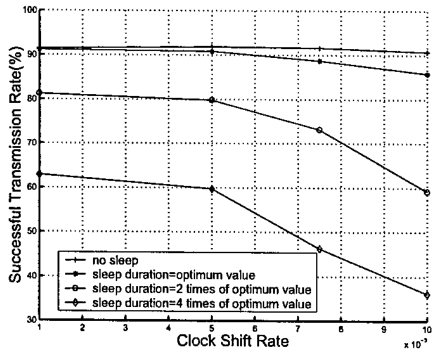


Fig. 10. Data Packet Successful Transmission Rate

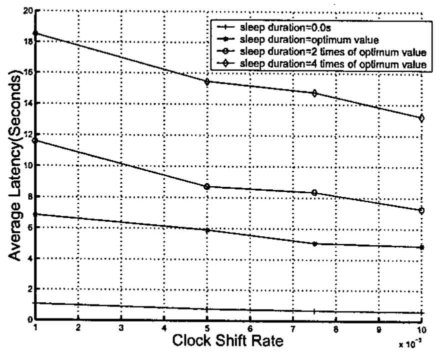


Fig. 11. Average Time Delay for Data Pacet

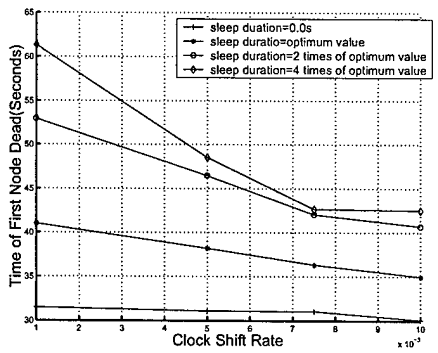


Fig. 12. Time of First Node Dead

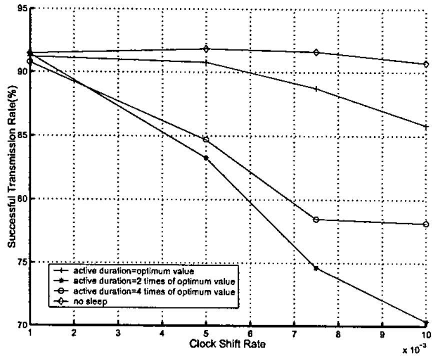


Fig. 13. Data Packet Successful Transmission Rate

contention-based and energy-efficient MAC protocols. We use schedule broadcast and clock drift aware scheme to dynamically solve the system schedule un-coincidence problem. In this paper, we also supply a parameter designing model to optimize key parameters. Moreover, we make theory analysis on average latency (Point-to-Point) for our algorithm. The simulation results prove that our algorithm can successfully obtain optimum re-schedule duty-cycle, power off duration and power on duration to achieve higher average successful transmission ratio, shorter data packet average time delay and longer network lifetime.

Acknowledgement

This work was supported by the U.S. Office of Naval Research (ONR) Young Investigator Program Award under Grant N00014-03-1-0466.

References

[1] C. K. Toh, " Ad hoc mobile wireless networks: protocols and systems ," *Prentice-Hall,* 2002

[2] V.Rajendran, K.Obraczka, J. Juslin, O. Koukousoula, " Energy-efficient, collision-free medium access control for wireless sensor networks ," *SenSys'03,* Novermber5-7,2003, Los Angeles, California, USA.

[3] W. Ye, J. Herdemann, D. Estin " An energy-efficient MAC protocol for wireless sensor networks ," *IEEE INFOCOM,* 2002.

[4] T. V. Dam, K. Langendoen, " Adaptive energy-efficient MAC protocol for wireless sensor networks ," *SenSys'03,* Novermber5-7,2003, Los Angeles, California, USA.

[5] S. Singh and C. Raghavendra, " PAMAS: Power aware multi-access protocol with signaling for ad hoc networks ," em ACM SIGCOMM Computer,

[6] K. Romer, " Time synchronization in ad hoc networks ," *Proceedings of MobiHoc,* 2001.

[7] W. Su and I. F. Akyildiz, " Time-diffusion synchronization protocol for sensor networks ," *Technical report,* Deorgia institute of technology, broadband and wireless networking laboratory, 2002.

[8] S.Ganeriwal, R.Kumar, S.Adlakha, and M,B.Srivastava, " Network-wide time synchronization in sensor networks ," *Technical report,* University of California, Dept. of electrical engineering, 2002.

[9] J. Elson and D. Estin, " Time synchronization for wireless sensor networks," *Technical report,* UCLA CS, 2002.

[10] " IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications ," Nov.1997. P802.11.

[11] M. M. Carvalho, J.J. Garcia-Luna-Aceves, " Delay analysis of IEEE 802.11 in single-hop networks ," *Proceedings of 11th IEEE international conference on network protocol(ICNP'03*

[12] D.Bersekas and R.Gallager, " Data Networks ," Englewood Cliffs, NJ: Prentice-Hall, 1987.

[13] L.H.Bao, J.J. Garcia-Luna-Aceves " Hybrid channel access scheduling in ad hoc networks ," *IEEE Computer Society,* Washington, DC, USA.

[14] M. Stemm and R.H.Katz " Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices. ," *IEICE Transactions on Communications,* E80-B(8): 1125-1131, 1997

[15] D.Culler, D.Estin and M.Srivastava " Overview of Sensor Networks. ," *IEEE Computer Society,* pp.41-pp.49, Aug. 2004

[16] M. J. Handy, M. Haase, D. Timmermann " Low energy adaptive clustering hierarchy with deterministic cluster-head selection ," Mobile and Wireless Communications Network, 2002. 4th International Workshop on, 9-11 Sept. 2002 Pages:368 - 372

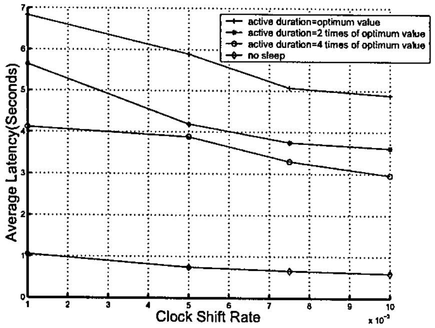


Fig. 14. Average Time Delay for Data Pacet

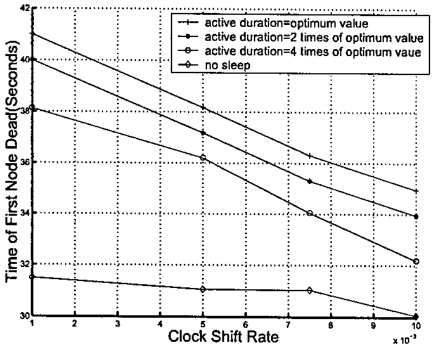


Fig. 15. Time of First Node Dead

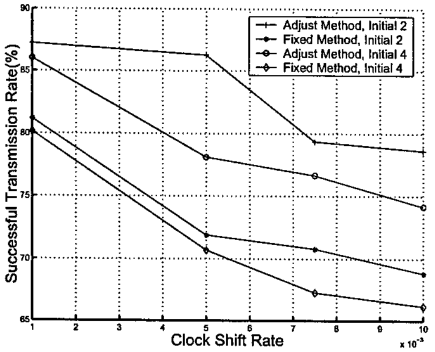


Fig. 16. Data Packet Successful Transmission Rate